21世纪高职高专规划教材

计算机应用系列

网站项目规划与设计

乔冰琴 主编 霍福华 张海玉 副主编

清华大学出版社

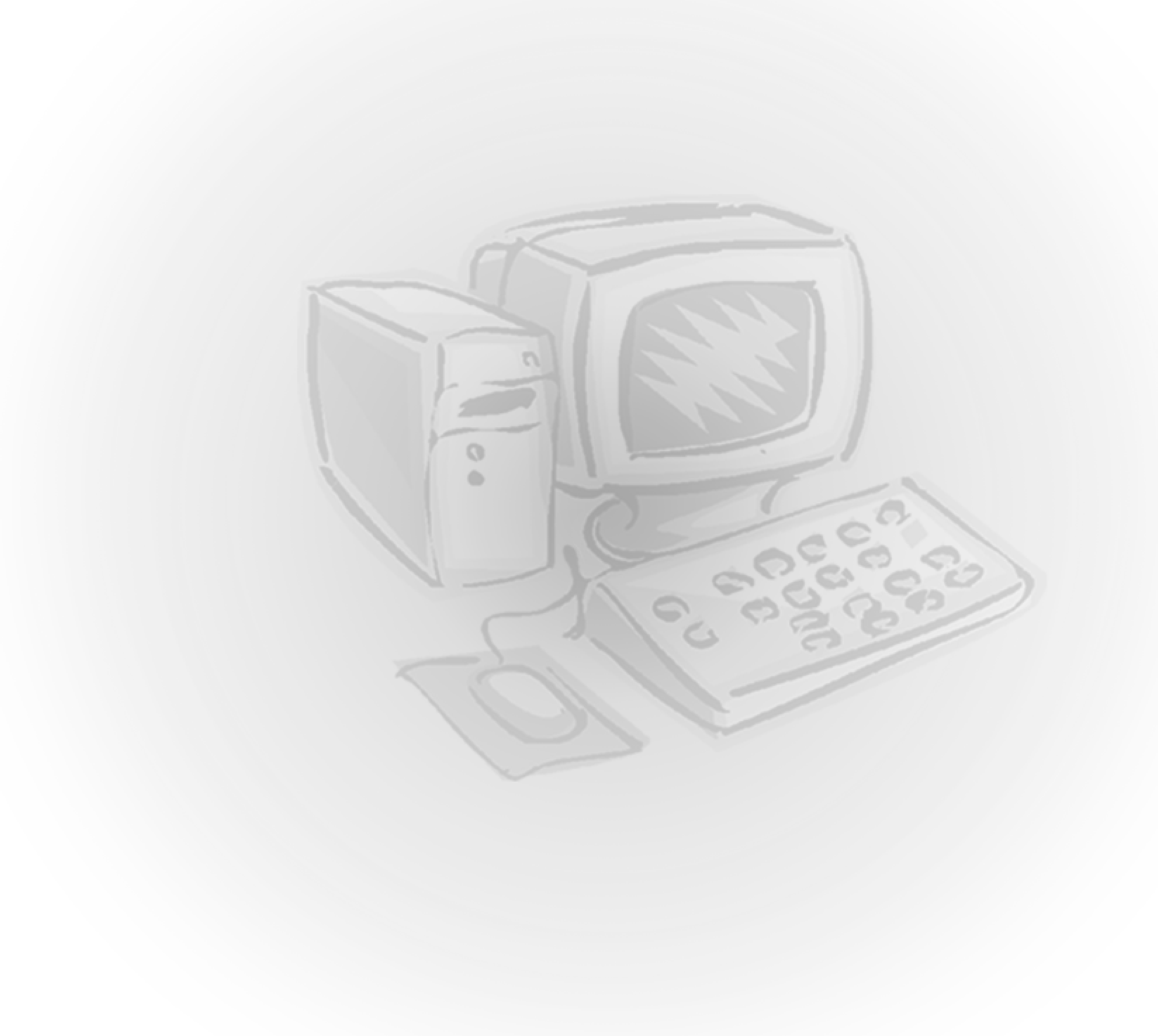
21世纪高职高专规划教材

计算机应用系列

网站项目规划与设计

乔冰琴　主编

霍福华　张海玉　副主编



清华大学出版社

北　京

内 容 简 介

本书以“新闻发布系统”网站项目为例，从网站的规划和设计着手，介绍了一个完整网站项目的开发过程。配合网站的开发过程，逐步引入目前流行的网站开发技术和开发环境，包括 XHTML 编码知识、CSS、JavaScript、JSP 等技术，并采用 Dreamweaver CS3 作为网站前台开发环境，Eclipse 作为服务器端 JSP 的开发环境，MS SQL Server 2005 作为数据库环境。同时介绍了支撑网站运行的环境搭建方案与其他支撑策略。

本书适合的读者对象包括高校中学习网页制作与网站建设的各专业学生，以及对网页制作和网站建设有兴趣的读者，尤其适合作为高校网站项目建设课程的实践性指导教材。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目（CIP）数据

网页项目规划与设计/乔冰琴主编. —北京：清华大学出版社，2009.11
21 世纪高职高专规划教材. 计算机应用系列
ISBN 978-7-302-20799-3

Ⅰ. 网…　Ⅱ. 乔…　Ⅲ. ①网站—项目管理—高等学校：技术学校—教材 ②网站—设计—高等学校：技术学校—教材　Ⅳ. F407.67 TP393.092

中国版本图书馆 CIP 数据核字（2009）第 154785 号

责任编辑：张龙卿
责任校对：刘　静
责任印制：

出版发行：清华大学出版社　　**地　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　销：全国新华书店
开　本：185×260　　**印　张**：22.25　　**字　数**：540 千字
版　次：2009 年 11 月第 1 版　　**印　次**：2009 年 11 月第1 次印刷
印　数：1～0000
定　价：00.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177 转 3103　　产品编号：031253-01



前　　言

一般在网站制作的初步教程中，都会讲到如何使用模板“迅速”创建站点。此后，“模板”一词便容易在读者的脑海里扎下根来。在日常的教学中，教师们也常教学生从网络里搜索漂亮的网站模板并进行修改，然后加入自己的内容，便形成学生自己的“作品”。不可否认，这是一种不错的教学方式，学生们能够迅速地从他们自己的学习成果里体会到学习的快乐。但是在教学的后期，也就是项目实战阶段，一定要给学生讲清楚，实用的网站既不能只靠模板来创建，也不能只是从网上搜集一些自己喜欢的素材，然后进行简单的堆砌进行创建，这样做是不可取的。

在学生们的学习中，还有一种现象，就是学习了网页制作课程后，基本能够按照老师的要求去完成简单的网站开发工作，或者通过模仿现成的网站来实现一些简单的功能，但是如果要求学生单独完成一个新网站的开发，一般都会觉得无从下手，这是由于教师在教学中没有把课本知识和实际工作要求很好地结合起来。

这么说有虽批评的意味，但我自己也是教师，也教授学生们网页制作，所以以上也是我对自己工作的总结。写作本书的目的，就是希望在学生们学完网页制作的基本知识后，教师能够带领大家从头到尾做一个网站，引导学生们学会如何将所学的零碎知识点贯穿到网站开发的各个工作过程中去。

本书的组织采用了国内先进的教学理念——“工作过程系统化”进行开发，并通过一个实际网站项目的完整开发过程进行介绍。本书的每一章都是动态网站开发的一个工作环节，每一节都是工作环节里的具体工作任务。全书对每个环节的工作任务都进行了描述，围绕此工作任务将教学内容分为知识储备(为完成这一任务所必须具备的前提知识与技能)、任务描述(本环节的任务要求)、任务实现(本任务的实现技术和步骤)、相关知识链接(与本任务有关的其他扩充性知识)、经验总结(完成本任务需要注意的事项及一些经验)。全书各章的“任务描述”部分形成了网站开发与运营维护的各个工作环节，“任务实现”部分对这些工作步骤做了具体的讲解。因此，如果通读本书的这两部分内容，就能够看到一个完整的网站开发与运行的全部过程，包括网站的设计、开发、推广、维护等过程。

本书选材合理，组织新颖，既有一定的网站开发理论知识，又能体现极强的工作实践要求。该网站采用 JSP + SQL Server 作为开发环境，就是考虑到 JSP 是跨平台的动态服务器端编程技术，适用范围比较广。

本书由乔冰琴主编，霍福华、张海玉担任副主编。第 1、3 章由山西财政税务专科学校乔冰琴编写，第 2、4、5、7、8 章由山西国际商务职业学院霍福华编写，第 6、9、10、11 章由山西财政税务专科学校张海玉编写。另外，杨雪、肖丹、孙启强、贾小平、李军、李婷、杨蕾、孙琪、吴建文、李钢、范荣等也参加了部分内容的编写及素材整理工作，在此一并表示感谢。



尽管书中可能存在这样或那样的缺陷和不足，但是作者非常希望大家学习完本课程并在实际工作中开发与维护网站时，能够更容易地胜任这些岗位的工作要求，顺利融入角色中。欢迎广大高校师生和网站开发工作者批评指正。欢迎大家到清华大学出版社网站www.tup.com.cn下载或给作者来信(qiaobingqin@sxftc.edu.cn)索取网站项目开发资料。

编　者

2009年6月



目　录



















第1章　网站需求分析

许多网友都有一些做网站的技术，能做一些看上去不错的网页。可是动手做一个由许多网页按一定逻辑组成的网站时，就不知从何处下手了。即使是一些站长，在做网站时也常会忽视一个重要的问题，那就是：网站最终的目标是什么？如果这个问题不明确，建议不要动手做任何网页，即使很清楚地知道如何在网站上做飘移广告或者做很漂亮的跟着鼠标跑的一串亮闪闪的小星星。

明确了网站需求，才能确定网站目标。这个目标必须是实在的、可以操作的、可以指导设计的，要能集所有网站运营行为而不是表面敷衍了事的。到底想让浏览者在网站上做什么？想让浏览者做哪件具体事情？简单地讲就是网站目标。但这个目标并不是不费任何力气就能弄清楚的，因为目标经常并不是一眼就能看出来。比如电子商务网站的目标，大家可能会不假思索地说就是让客户买东西。但实际上是这样吗？不一定。如果网站是运用电子邮件作为网络营销的主要手段，那网站的目标可能就是想办法收集客户的电子邮件地址，而不是直接在网站上向客户销售商品。

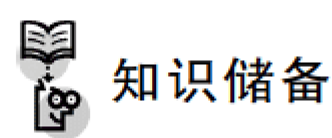
雷波特（Rayport，美）和贾沃斯基（Jaworski，美）在2001年提出被他们命名为7Cs的有效网站的必要设计元素，分别是组织（布局与设计）、内容（文字、图片、声音和录像）、社区（网站如何实现访问者之间的交流）、专门化（网站满足不同访问者要求的能力以及提供个性化服务的能力）、沟通（如何实现网站与访问者之间的双向传播）、联系（网站与其他网站的联系程度）以及商务（网站进行商务交易的能力）。

需要注意的是，不论是以提供信息为主要特征的信息类型网站，还是以电子商务交易为主的交易类型网站，或者是以游戏为主的游戏网站，或者以提供某种特定功能服务的功能类型网站，或者是综合上述内容的综合类型网站，在其开发之前，都必须要做一项重要的工作，那就是网站要为浏览者提供的网站基本内容是什么？能让浏览者使用网站达到什么样的目的，即网站的终极目标是什么？

网站建设在网站需求分析阶段的工作任务包括：确定网站类型、确定网站目标、确定网站主题、确定网站的CI形象、确定网站的功能和性能、确定网站的系统配置、确定网站的风格和创意、确定网站结构、搜集素材。



1.1 确定网站类型


知识储备

(1) 网站是一个具有大量信息的综合体，是由许多张网页组成的，这些网页按特定的要求进行链接，形成一个既有鲜明风格又有完整内容的有机整体。

(2) 网站需求分析工作就是在动手写网页代码之前，通过一些调查或者其他的手段，弄清楚网站目标，以及达到这个目标都需要做哪些工作，让整个网站开发小组都明确地知道网站要完成哪些功能，网站主题是什么，网站的风格和网站结构是什么。

1.1.1 任务描述

确定“新闻类别网站”的网站类型。

1.1.2 任务实现

“新闻类别网站”主要用于信息的及时发布，并对信息进行综合管理，这种系统就是CMS，CMS是Content Manager System的缩写，译为新闻发布系统或信息发布系统。目前，许多网站都包含新闻发布的功能。

1.1.3 相关知识拓展

网站类型的划分可以按照不同的方法来进行，划分标准不同，可以有不同的划分结果。我们可以按照行业分、按照区域分、按照语言分、按照意识分，还可以按照其他的类型来分。网站的划分标准和对应的划分结果不需要掌握，需要知道的是一般常见的网站类型都有哪些。

下面给出的划分结果没有什么权威性，只是列出一些常见的网站类型名称以供大家参考。

1. 第一种划分结果

第一种划分结果为：企业内部管理网站、宣传网站、实验网站、门户网站、交易网站(例如网上超市、网上专卖店、游戏网站、提供特殊交易的网站)、中介网站(例如网上商城、网上拍卖、网上提供信息服务)、行业网站(例如网上学校、网上旅游)、电子政务网站。在这种划分结果下，“新闻类别网站”多属于上述网站中的一个版块。

2. 第二种划分结果

第二种划分结果为：展示型网站、内容型网站、电子商务型网站、门户型网站。

- 展示型网站：网站主要以展示形象为主，艺术设计成分比较高，文字内容不多，多见



于从事美术设计的工作室网站。
- 内容型网站：网站以内容为重点，用内容吸引人。例如，普通的公司网站，用于发布公司产品、公司动态、招聘信息等。一些从事信息服务性的站点，如文学站、下载站、新闻站等。这类站点的设计一般以简洁大方为主，不需要太多太花哨的东西转移浏览者的视线。
- 电子商务型网站：网站以从事电子商务为主，站点对安全性和稳定性要求很高。这类站点设计一般要求简洁大方，但又不失热闹，能聚集人气，网站颜色多用蓝色等能表现信任感的色彩。
- 门户型网站：这类站点类似内容型网站，但又不同于内容型网站。门户型网站的内容特别丰富，内容比较综合；而内容型网站的内容一般较集中于某一专业，或所属的领域，或所属的公司，或工作室等小范围的内容。另外门户型网站除了表现更为丰富的内容外，还特别注重网站与访问者之间的交流，门户型网站会提供信息的发布平台及与用户的交流平台等。

在这种划分结果下，“新闻类别网站”应属于内容型网站。

3. 第三种划分结果

第三种划分结果为：单纯广告网站、垃圾网站、产品网站、正规网站。
- 单纯广告网站：就是纯粹广告代码页面，即纯广告页的形式。这种网站处在最基本的网页阶段，网站的特点是靠宣传流量带来收入。这是最常见的一种广告形式。
- 垃圾网站：垃圾网站的赢利途径是靠广告联盟的广告，垃圾网站通常是采取一些比较有诱惑力的图片或者文字，目的是吸引住一些自然流量或者是利用搜索引擎获得一些流量。例如小电影站，以及一些美女图片站，都属于垃圾网站。
- 产品网站：这类网站一般都不带有各类广告，其主要的赢利模式是靠销售产品来获得利润，例如企业电子商务网站。
- 正规网站：正规网站的定位具有一定的创新性和独立性，能够在所属市场中占领一定的份额，或者是目前市场还属于空白的，或者是处于绝对的垄断状态。正规网站建立的前提是站长对网站所属的市场非常了解，这类网站一般都是靠网站的增值服务或者是特殊服务来赢利的。

在这种划分结果下，“新闻类别网站”属于上述产品网站和正规网站中的一个版块。

4. 第四种划分结果

第四种划分结果为：信息型、线上销售型、线下销售型、提升知名度型、资源型等。
- 信息型：提供给别人尽可能多的某方面的信息。
- 线上销售型：即电子商务类。
- 线下销售型：网站用来提供产品的信息，实际产品在线下销售。
- 提升知名度型：艺术设计成分比较高，但内容不多，多见于从事美术设计方面的工作室网站。
- 资源型：提供资源让别人下载。

还有媒体型、教育型、娱乐休闲型……



？小思考

在这种划分结果下，“新闻类别网站”属于哪种类型呢？

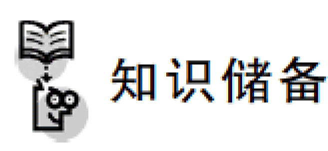
1.1.4 经验总结

在网站需求分析工作过程中，先根据对网站相关信息的基本了解，确定网站属于哪种类型，其目的是为了确定网站的大方向，也就是确定网站的边界，即哪些内容属于网站，哪些内容不属于网站。

网络时代大家都想靠做网站获得利润，那做什么类型的网站能赚钱呢？大家可能会发现，现在网络上赢利的网站类型主要有单纯广告网站、企业的产品网站、提供特殊服务的网站，还有众人比较讨厌的垃圾网站。网站不一定都能赢利，单纯广告网站一天也有收入几万元的，但也有建设很好的正规网站一直处于亏损状态；有做垃圾网站发财的，但也有做产品网站赔本的。按照市场需求来正确定位网站类型，可能是通往成功的第一步。

注意确定网站所面临的目标访问群，目标访问群的需求与定位网站类型以及网站主题、风格等都有直接的关系。

1.2 确定网站主题和目标



知识储备

所谓主题，也就是网站的题材。网络上的网站题材千奇百怪，琳琅满目，只要想得到，就可以把它制作出来。下面列出美国《个人计算机》(PC Magazine)杂志评出的某年度排名前100位的全美知名网站的十类题材，具有一些参考价值。

①网上求职；②网上聊天/即时信息/ICQ(I see you，中文意为“我在找你”，俗称网络寻呼机，实时通信软件)；③网上社区/讨论/邮件列表；④计算机技术；⑤网页/网站开发；⑥娱乐网站；⑦旅行；⑧参考/资讯；⑨家庭/教育；⑩生活/时尚。

这十个大类都可以继续细分，比如第六类娱乐网站可以继续细分为体育、电影、音乐类；音乐类还可以按格式分为MP3类、VQF类、Ra类，也可以按表现形式分为古典类、现代类、摇滚类等。网站题材除了上述最常见的主题外，还有许多专业的、另类的、独特的主题，比如心理咨询、数控设备、天气预报等。同时，各个题材相联系和交叉结合又可以产生新的题材，例如旅游论坛(旅游+讨论)、经典足球播放(足球+影视)等。

1.2.1 任务描述

确定“新闻类别网站”的主题和目标。

1.2.2 任务实现

“新闻类别网站”主要用于新闻信息的及时发布，并进行新闻信息的综合管理，网站的主题和目标定位在“红色革命”题材方面的新闻。





1.2.3　相关知识拓展

1. 选择题材

在制作网站之前，必须要考虑选择什么样的网站主题。对于题材的选择，建议从以下几方面考虑。

（1）主题要小而精。主题小而精是指网站主题的定位要小、要精。如果想制作一个包罗万象的站点，如果尝试着把所有精彩的东西都放到网站上去（但这确实是初学者喜欢做的事，初学者对于喜欢的东西总是不舍得删掉），那么往往会事与愿违，给人的感觉是没有主题、没有特色，面面俱到却都很肤浅。一般的网站制作者没有也不可能有足够的时间和精力去制作一个这样的站点。作者在教授网页制作的过程中，一眼就能看出自己的学生是不是初学网站制作的，学生们总是从网上搜集许多喜欢的文字、图片、Flash 动画、视频等，然后不加区分地堆积在一起，整体感觉是网站上的内容东一榔头西一棒子。一定切记，网站不能没有主题、没有特色，不能好像什么都有却都很肤浅。

网络的最大特点就是新和快，目前最热门的个人主页都是天天更新，甚至几小时更新一次。最新的调查结果也显示，网络上的"主题站"比"万全站"更受人们喜爱，就好比专卖店和百货商店，如果需要买某方面的东西，多会选择专卖店。再举个例子，作者的一位网友希望制作文学方面的题材，但是文学也包括多方面的内容，有小说、诗歌、散文，有科幻、武侠、推理，作者在仔细了解了网友的擅长和想要提供的主要内容之后，将题材定位在新兴的网络文学上，删除了原有的一些无关的文学作品，集中扩大和整理网络文学作品。网站推出后，很快受到众网友的喜爱。

网站主题小并不等价于网站内容少，网站甚至可以只有一个方面的内容，但这一方面的内容一定要比较全。

（2）题材最好是自己擅长或者喜爱的内容。在网站内容的设计上可以从自己的专业或兴趣爱好方面多做考虑，比如：若擅长编程，就可以建立一个编程爱好者网站；若对足球感兴趣，就可以报道最新的战况、球星动态；若在流行音乐、数控、书法、绘画等某方面有独到的功夫，就可以将此专题作为网站内容，这样在制作时，才不会觉得无聊或者力不从心。一定要记住兴趣是制作网站的动力，没有热情，很难设计制作出杰出的作品。

2. 确定网站目标

在确定了网站的主题之后，就必须要考虑网站的目标了。作为网站的设计者，若在设计网站时不清楚让浏览者浏览网站时做什么，那么当浏览者实际浏览网站时，也自然不知道要干什么。网站目标的确定建议从以下几方面考虑。

（1）网站目标不要太高。"目标太高"是指在这一题材上已经有非常优秀、知名度很高的站点，不要把网站目标定位于超过这样的站点，因为超过是很困难的。除非能下决心和有实力竞争并超过它。但同时要记住，在互联网上只有第一，人们往往只记得最好的网站，第二名、第三名的印象则会浅得多。

（2）想知道网站目标的人可能有不同的角色，如网站的客户、网站的老板和站长等。给



站长看的网站目标一定要非常明确,尽管有时候想非常明确地定义网站目标不是很容易的事。网站目标不能简单写成:“最大限度方便客户”或者“给客户提供最大价值”之类,这是写给客户看的;也不能写成:“促进公司销售增长10%”或者“极大地提高企业知名度”,这是写给老板看的。上述的网站目标描述太宽泛,对于站长来说无法指导网站的开发过程。要是认为网站目标就是赚钱,那可能会导致无法开始进行网站的开发。

(3) 应注意网站目标不能包括很多要求,否则在设计网站时,各个要求可能会产生冲突,使得在实现时难以抉择,也会使得浏览者不能很好地在网站里按站长的想法进行操作,从而难以达到预定的网站目标。单一目标最有利于网站实现并且最有利于达到建站的最终目标,即使难以将网站目标描述成一个单一目标,也建议大家最多为单一目标增加一个辅助目标,但千万不要面面俱到。

定位网站主题和目标是设计网站的第一步,也是很重要的一部分。如果能找到一个合适的主题和目标,多花一些时间做这件事是值得的。

(4) 不同的网站目标,在进行网站设计、内容写作、实现流程安排上都需要有不同的处理。

如果网站是依靠广告来赢利,不同的广告方式会形成不同的网站目标;如果网站是按单击量收费,那么网站目标就是让浏览者来到网站后尽快单击广告,然后离开;如果广告是按显示次数收费,策略就正相反,浏览者在网站上停留时间越长越好,看网站上不同的网页越多越好。

如果网站是按行动收费,或参加网站联盟,表面看起来要做的也是让浏览者尽快单击链接,然后离开。但其实不然,网站必须尽一切努力,让浏览者还没有离开网站时,就已经决定要买网站推荐的产品了。这种情况下,网站就要有一些产品推荐、产品使用经验等文字,而不仅仅是放一个旗帜广告或链接那么简单。

(5) 网站目标必须是明确而单纯的,或者尽可能让浏览者在网站内逗留,或者尽快离开,或者留下电子邮件地址,或者让浏览者尽快打电话给网站服务人员,或者下载试用软件。

总之,不同的网站赢利模式会产生不同的网站目标,网站目标又反过来指导网站设计。

1.2.4 经验总结

1. 网站主题要与网站内容相符

对于个人网站,必须要找准一个自己最感兴趣的内容来做深、做透,办出自己的特色,这样才能给网站浏览者留下深刻的印象。网站的主题无定则,只要是感兴趣的,任何内容都可以,但主题一定要鲜明,在主题范围内可将网站内容做得大而全、精而深。

2. 网站起名要遵循的原则

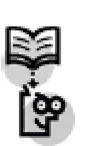
许多人可能认为网站名称与网站设计无关,其实网站名称也是网站设计的一部分,而且是很关键的一个要素。和现实生活中一样,网站名称是否正气、响亮、易记,对网站的形象和宣传推广也有很大影响。下面是一些建议(资料来源:http://www.pconline.com.cn/pcedu/sj/wz/other/0404/363725.html):

(1) 名称要正。名称要合法、合情、合理。不能用反动的、色情的、迷信的、危害社会安全的名字。

(2) 名称要易记。根据中文网站浏览者的特点，除非有特定需要，网站名称最好用中文名称，不要使用英文或者中英文混合型名称。网站名称的字数应该控制在六个字(最好四个字)以内(网站名称字数少还有个好处，一般友情链接的小 Logo 尺寸是 88×31(单位为 pixels，即像素)，而六个字的宽度是 78pixels 左右，适合于其他站点的链接排版)。

(3) 名称要有特色。名称朴实就可以接受，如果能体现一定的内涵，给浏览者更多的视觉冲击和空间想象力，则为上品。网站名称在体现出网站主题的同时，最好能点出其特色之处。

1.3　确定网站的 CI 形象



知识储备

网站树立 CI 形象的关键是做好网站标志、色彩、字体、标语这四个方面的工作。

1. 设计网站的标志

需要为网站设计制作一个网站标志(Logo)。如同商标一样，Logo 是站点特色和内涵的集中体现，看见 Logo 就让大家想到是哪个站点的标志。注意：这里的 Logo 不是指 88×31 的小图标 banner，而是网站的标志。

标志可以是中文、英文字母，可以是符号、图案，也可以是动物或者人物等。例如：www.google.cn 是用 google(Google)的英文作为标志，www.sina.com 用字母 sina＋眼睛(sina)作为标志。

标志的设计创意来自网站的名称和内容。例如：若网站有代表性的人物、动物、花草，可以用这些作为设计的蓝本，加以卡通化和艺术化，百度的百度熊( )、搜狐的卡通狐狸( )、土豆网的笑脸( )都是这样的范例。

若网站是专业性的，可以用本专业有代表性的物品作为标志。比如中国银行的铜板标志( )，奔驰汽车的方向盘标志( )。

用网站的英文名称做网站的标志是最常用和最简单的网站标志设计方法，对英文名称采用不同的字体来表现，或者对字母进行变形，都可以较容易地制作好网站的标志。

2. 设计网站的标准色彩

网站给人的第一印象来自视觉冲击，确定网站的标准色彩是相当重要的一步。不同的色彩搭配产生不同的视觉效果，并可能影响到访问者的情绪。

“标准色彩”是指能体现网站形象和延伸内涵的色彩。例如：IBM 的深蓝色，肯德基的红色条型，Windows 视窗标志上的红蓝黄绿色块，都使人觉得很贴切，很和谐。

一个网站的标准色彩一般不超过 3 种，太多则让人眼花缭乱。标准色彩要用于网站的标志、标题、主菜单和主色块，给人以整体统一的感觉。除了标准色，其他色彩也可以使用，但建议只将其他色作为点缀和衬托，不要喧宾夺主。

适合于网页标准色的颜色主要有蓝色、黄/橙色、黑/灰/白色三大系列色。



3. 设计网站的标准字体

和标准色彩一样，标准字体是指用于标志、标题、主菜单的特有字体。一般情况下网页默认的字体是宋体。为了体现站点的“与众不同”和特有风格，可以根据需要选择一些特别字体。例如，为了体现专业可以使用粗仿宋体，体现设计精美可以用广告体，体现亲切随意可以用手写体等。要根据网站所表达的内涵，选择更贴切的字体。使用字体时，可以从目前常见的中文字体和英文字体中进行选择，常见的中文字体有二三十种之多，常见的英文字体更是有近百种之多，还可以从网络上下载专用英文艺术字体。总之，字体是很多的，关键是选择合适的、满意的字体。

4. 设计网站的宣传标语

网站的宣传标语是对网站精神、网站目标的高度概括和宣传。

1.3.1 任务描述

确定“新闻类别网站”的整体形象(CI 形象)。

1.3.2 任务实现

“新闻类别网站”是以“红色革命”为题材的，所以网站的整体形象以红色为主色调，配以灰色和白色，Logo 直接以“红色新闻网”体现，具体设计可自主处理，整个站点的字体采用 verdana、arial、helvetica、sans-serif 字符集。

1.3.3 相关知识拓展

网站的 CI，是借用的广告术语。CI 是英文 Corporate Identity(企业视觉形象识别)的缩写，意思是采取各种措施塑造企业形象的过程，通过视觉来统一企业的形象。现实生活中的 CI 策划比比皆是，例如可口可乐公司，设计有全球统一的标志、色彩和产品包装，给人们的印象极为深刻。

一个杰出的网站，像实体公司一样，也需要整体的形象包装和设计。既准确又有创意的 CI 设计，对网站的宣传推广有事半功倍的效果。在网站主题和名称定下来之后，需要思考的就是网站的 CI 形象。

许多人自己不是学广告专业的，觉得网站的 CI 做起来可能太难。如果是为已有的企业产品做网站，网站的 CI 要与企业的产品形象一致。如果没有现成的企业形象，CI 还是需要网站设计者来做。但大多数网页设计师都没有系统地学过广告设计，所以最好参考一些其他网站的具体做法，加上自己的个人经验，再确定自己网站的 CI。

1.3.4 经验总结

一个优秀网站都有自己的个性和文化，而且会从一开始就千方百计去体现它，树立起属于自己的风格。





在浏览网站时，站点的整体形象给浏览者的综合感受属于网站风格的展现。网站的整体形象受站点的 CI（标志、色彩、字体、标语）、版面布局、浏览方式、交互性、文字、语气、内容价值、存在意义、站点荣誉等诸多因素的影响。

网站的风格可以在色彩、技术上着重体现，以使浏览者只看到网站的一部分，就能明确分辨出这是哪个网站。网站风格还可以通过网站的外表、内容、方字和交流方式体现出站点的个性与情绪，是温文尔雅的还是执著热情的，是活泼易变的还是放任不羁的。

做网站时，不要将网站建成一个堆砌在一起的信息的集合。向别人描述网站时，不能只用专业的词语来描述，诸如信息量、浏览速度之类。好的网站在浏览过一次后就能让人“过目不忘”，下次还来。

所以，要让网站“与众不同”。

在网站字体方面需要注意的是：使用非默认字体时只能用图片的形式，因为很可能浏览者的计算机里没有安装这种特别字体，如果真是如此，那么为网站辛苦设计制作的效果便付之东流了。

小贴士

有许多专业 Logo 设计网站提供了 Logo 的设计与制作服务，从这些网站上可以学到和 Logo 制作有关的知识，也可以欣赏到许多优秀的 Logo，建议去这些网站看看。

1.4　确定网站的功能和性能



知识储备

网站功能需求定义，是指明确网站必须具有什么功能才能达到网站的目标？这个目标必须让所有的网站开发参与者做到心中有数。

网站性能需求定义，是指网站运行的稳定性、页面下载速度的快慢、界面是否友好、运行时间能不能满足需要、安全性能不能得到保证等因素。

具体地说，确定网站的功能和性能可以遵循下面的简单步骤来进行：

（1）与网站未来的主要使用者沟通，确定网站的功能，并列出所确定的网站功能的具体描述。

（2）确定网站的性能需求，并列出所确定的网站性能的具体描述。

（3）列出网站的其他需求描述。

1.4.1　任务描述

完成“新闻类别网站”的功能需求定义和性能需求定义。

1.4.2　任务实现

具体操作步骤如下：

（1）“新闻类别网站”的功能需求

“新闻类别网站”的系统功能模块设置如表 1-1 所示。



表 1-1 系统模块设置

编号	模块名称	简要描述
1	用户管理模块	网站用户注册、登录,修改个人资料
2	栏目模块	前台分类列表显示栏目,后台可以增加、删除、修改栏目信息
3	专题模块	前台分类列表显示专题,后台可以增加、删除、修改专题信息
4	新闻模块	前台显示新闻,后台可以增加、删除、修改新闻信息
5	友情链接模块	前台分类显示链接对象,后台对友情链接进行增加、删除、修改操作
6	系统日志管理模块	系统自动记录网站操作日志,后台可以查看
7	系统配置管理	记录系统的一些配置信息及网站数据统计结果,网站可以显示这些信息

各模块详细功能列表如表 1-2 所示。

表 1-2 各模块详细功能列表

编号	功能名称	模块编号	功能描述	输入内容	输出内容
1.1	用户登录	1	保证网站的合法用户通过身份验证进入网站操作	用户名、密码、验证码	用户登录状态
1.2	用户注册	1	提供网站浏览者注册成为本网站的正式用户	用户名、密码、性别、E-mail、电话等信息	注册的结果(提示用户“注册成功”或者“注册失败”)
1.3	修改个人信息	1	网站用户修改个人注册信息	修改的信息	提示修改的结果(“修改成功”或者“修改失败”)
1.4	后台登录	1	超级管理员、管理员登录后台管理界面	管理员名、密码	提示后台登录是否成功
2.1	添加栏目	2	管理员添加栏目	栏目相关信息	栏目列表
2.2	编辑栏目	2	管理员修改栏目信息	栏目相关信息	提示是否编辑成功
2.3	删除栏目	2	管理员删除栏目	选择欲删除的栏目	提示栏目删除成功还是失败
3.1	添加专题	3	管理员添加专题	专题相关信息	专题列表
3.2	编辑专题	3	管理员修改专题信息	专题相关信息	提示是否编辑成功
3.3	删除专题	3	管理员删除专题	选择欲删除的专题	提示专题删除成功还是失败
4.1	添加新闻	4	管理员添加新闻	新闻相关信息	新闻列表
4.2	编辑新闻	4	管理员修改新闻信息	新闻相关信息	提示是否编辑成功
4.3	删除新闻	4	管理员删除新闻	选择欲删除的新闻	提示新闻删除成功还是失败
4.4	浏览新闻	4	用户可以浏览任意一个栏目或者专题的新闻	单击任一条新闻	所选新闻的内容及相关新闻标题
5.1	添加友情链接	5	管理员添加友情链接相关信息	友情链接相关信息	友情链接列表

续表

编号	功能名称	模块编号	功 能 描 述	输 入 内 容	输 出 内 容
5.2	编辑友情链接	5	管理员修改友情链接相关信息	友情链接相关信息	提示是否编辑成功
5.3	删除友情链接	5	管理员删除友情链接相关信息	选择欲删除的友情链接	提示删除成功还是失败
6.1	记录日志操作	6	系统自动添加日志操作相关信息		日志操作列表
6.2	查看日志记录	6	管理员查看日志操作相关信息	单击任一条日志记录	所选日志的内容
7.1	记录系统配置信息及统计信息	7	管理员记录系统配置信息及统计相关信息	系统配置信息及统计要求	配置信息及统计结果
7.2	查看日志记录	7	管理员查看日志操作相关信息	选择欲查看的日志记录	所选日志的内容

(2)“新闻类别网站”的性能需求

① 时间需求

查询新闻的最长等待时间不超过 5 秒;更新信息的时间不超过 3 秒;数据上传和下载的时间不超过 10 秒。

② 空间需求

网站的硬盘空间预计为 1GB,内存空间预计为 512MB。

③ 其他需求

支持的终端数:<=1000。

(3)“新闻类别网站”的其他需求

① 界面需求

“新闻类别网站”的界面遵循方便、简洁、美观、大方及用户操作性强的宗旨。

② 安全性

网站浏览者只具有浏览新闻的功能,没有修改、删除新闻信息的权限;栏目、专题、友情链接只有管理员登录后台系统后才能实现管理;超级管理员对网站的操作者和访问者进行身份的验证和管理;任何等级的访问者登录时需要填写正确的验证码。

③ 可靠性

设计过程充分考虑恶意代码等非法入侵行为,尽量使网站安全可靠地运行。

1.4.3　相关知识拓展

需求分析是对网站的管理者、操作者和访问者进行访问调查,确定他们对目标网站的要求,这个过程看似简单,实则很难,而且在整个网站开发过程中相当重要。

实际工作中,网站操作者对网站的需求在整个开发过程中一成不变的很少。一般而言,由于他们对未知网站系统的模糊认识以及对业务流程表达得不规范、不清楚、不明确等问题的存在,网站开发人员很难从网站使用者那里得到他们对网站的准确完整的需求描述。



1.4.4 经验总结

现如今的网站尤其是企业网站，其功能可能很复杂，在动手进行网站的代码编写之前，一定要弄清楚网站最主要的功能需求和性能需求。即使是一个人开发网站，也至少需要网站未来的主要浏览者参与。除非是根据自己的需求给自己开发网站，这有可能是开发者一个人单干。请记住网站开发一定要考虑网站浏览者的需求。有许多专门开发网站的公司，常是一个网站开发的团队，此时在团队动手做网页之前一定要先让团队的全体成员搞清楚网站要完成的任务是什么，这个任务简单地讲就是网站的功能描述和性能描述。

在任何一个软件的开发中，软件的功能定义都是其中最难的。虽然有许多的开发模型可以规范软件开发过程，但开发的产品却不一定很好。要记住在得到网站的功能框架之前，要用心地和网站的未来访问者进行沟通。

1.5 确定网站结构

1.5.1 任务描述

完成"新闻类别网站"的网站结构定义。网站结构主要指网站的目录结构和链接结构。

1.5.2 任务实现

完成"新闻类别网站"的网站结构定义工作包括两个任务：

任务 1 确定网站的目录结构

任务 2 确定网站的链接结构

任务 1：确定网站的目录结构

"新闻类别网站"的目录结构如图 1-1 所示。

任务 2：确定网站的链接结构

"新闻类别网站"的链接结构如图 1-2 所示。

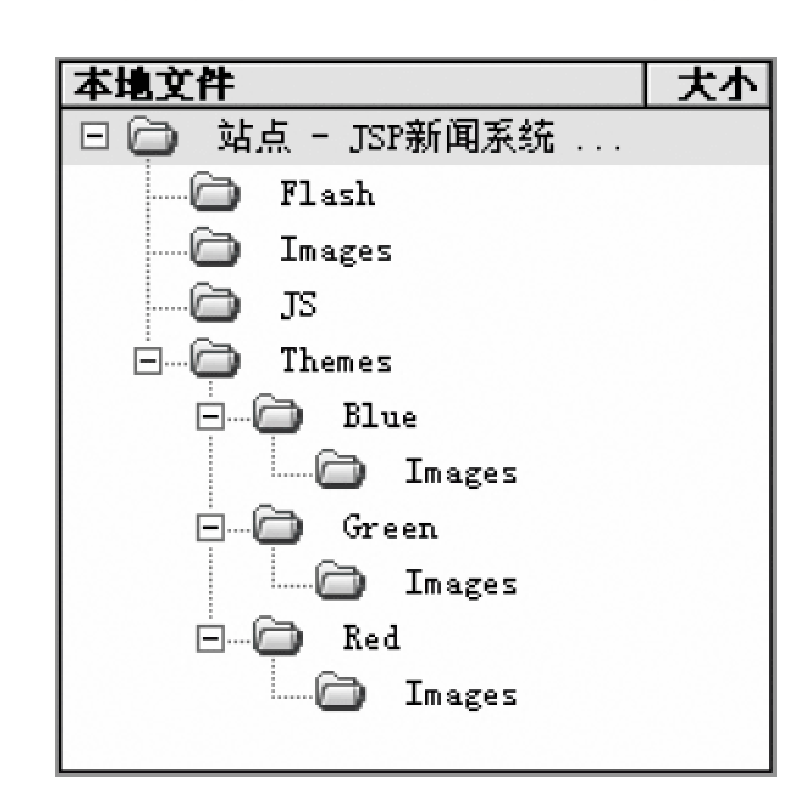


图 1-1 "新闻类别网站"的目录结构图

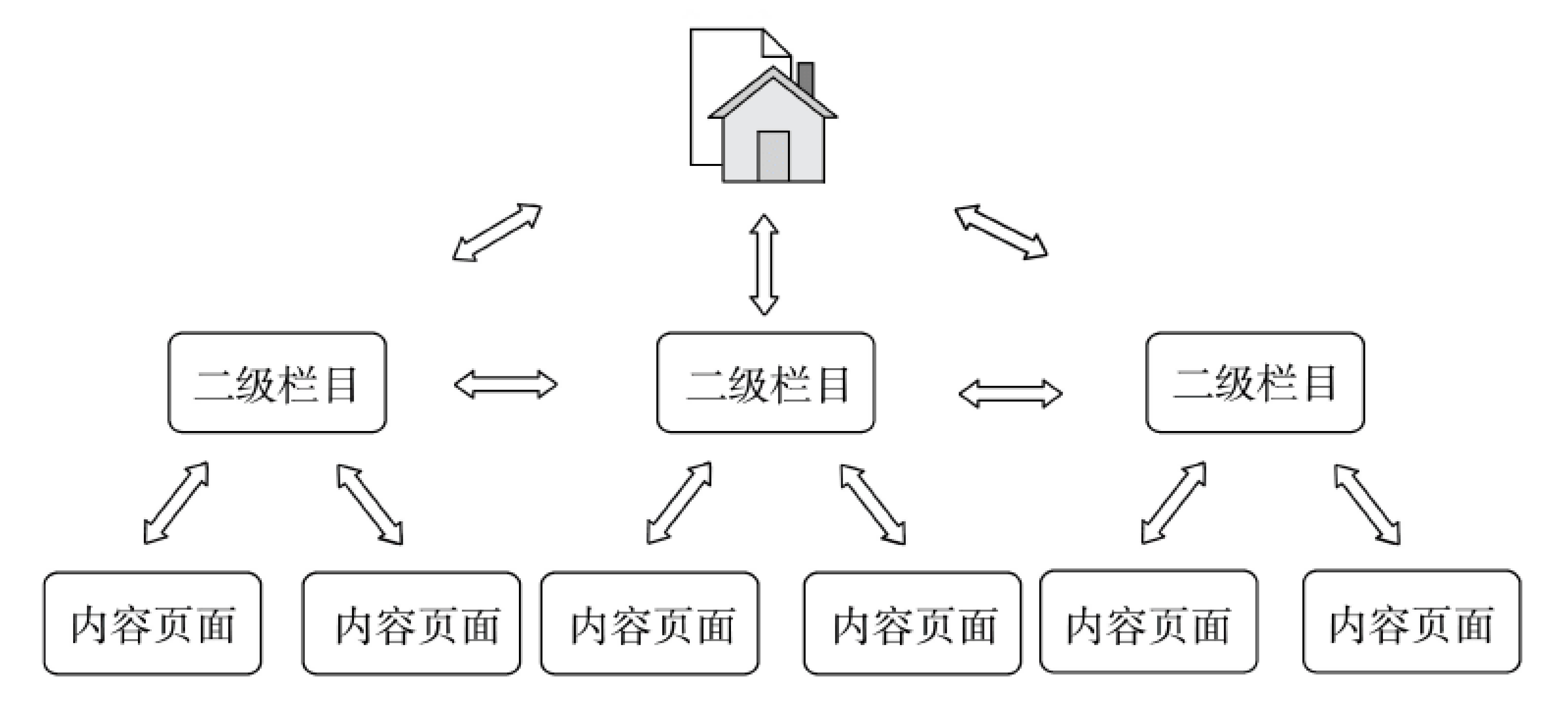


图 1-2 "新闻类别网站"的链接结构图



1.5.3　相关知识拓展

1. 网站的目录结构

网站的目录是指建立网站时创建的文件存取目录。网站目录结构是一个容易被忽略的问题，为数不少的站长也会不经规划，随意创建子目录。目录结构的好坏，对网站浏览者来说并不重要，但是对于站点本身的上传工作与运行中的维护工作、网站未来的内容扩充和网站移植都有着重要的影响。

2. 网站的链接结构

网站的链接结构是指页面之间相互链接的拓扑结构。它建立在目录结构基础之上，但可以跨越目录。形象地说，每个页面都是一个固定点，链接则是在两个固定点之间的连线。一个点可以和一个点连接，也可以和多个点连接。更重要的是，这些点并不是分布在一个平面上，而是存在于一个立体的空间中。

关于链接结构的设计，在实际的网页制作中是非常重要的一环。采用什么样的链接结构直接影响到网站版面的布局。例如主菜单放在什么位置，是否每页都需要放置，是否需要用分帧框架，是否需要加入返回首页的链接。在链接结构确定后，再开始考虑链接的效果和形式，是采用下拉表单，还是用 DHTML 动态菜单等。

随着电子商务的推广，网站之间的竞争越来越激烈，对链接结构设计的要求已经不仅仅局限于可以方便快速地浏览网页，而且更加注重链接的个性化和相关性。各大网站在显示某一主题的网页时，都会在页面周围添加与这一主题相关的链接，合理地利用这种网页链接方式能更有效、更持久、有更多机会留住访问者的眼球。

1.5.4　经验总结

1. 关于建立目录结构的一些建议

(1) 不要将所有文件都存放在根目录下。往往初学者为了方便，会将网站的所有文件都放在根目录下，这样做会造成文件管理混乱和上传速度慢等缺点。

(2) 按栏目内容建立子目录。子目录的建立，推荐按主菜单栏目建立。对于网站的次要栏目，类似友情链接等内容或者需要经常更新的栏目，可以建立独立的子目录；而不需要经常更新的栏目，可以合并放在一个统一目录下。网站的程序一般存放在特定目录下，以便于维护管理。网站提供的下载内容也可以放在一个目录下。

(3) 在每个主目录下都建立独立的 images 目录。通常，网页制作工具软件会在每个站点根目录下建立一个 images 目录。作者在刚开始学习网页制作时，习惯将所有图片都存放在这个默认目录里。可是后来发现这样做并不方便，当需要将某个主栏目打包供浏览者下载时，或者将某个栏目删除时，与这个栏目有关的图片管理就变得相当麻烦。经过实践发现，为每个主栏目建立一个独立的 images 目录是最方便管理的。而根目录下的 images 目



录可以用来放首页和一些次要栏目的图片。

(4) 目录的层次不要太多。目录的层次建议不要超过3层，这样维护管理比较方便。

(5) 不要使用中文目录。网络无国界，使用中文目录可能对网址的正确显示造成困难。

(6) 不要使用过长的目录。尽管服务器支持长文件名，但是太长的目录名不便于记忆。

(7) 尽量使用意义明确的目录。可以用Flash、DHTML、JavaScript、Images来建立目录，不要使用过于简单、不明其意的名称建立目录。

2. 网站的链接结构

尽量使用最少的链接，达到最有效率的浏览。一般而言，建立网站的链接结构有两种基本方式。

(1) 树状链接结构(一对一)

如图1-3所示，状似企业的组织机构图。网站首页指向一级页面，一级页面指向二级页面，以次类推。按照这样的链接结构浏览网页时，访问者从首页一级级进入下级网页，再从下级网页一级级退出直至主页。这种链接结构的优点是条理清晰，访问者能够明确地知道自己处于网站的什么位置，不会"迷失"于网站中；缺点是浏览效率低，比如从图1-3的二级页面A到二级页面B，必须绕经网站首页。

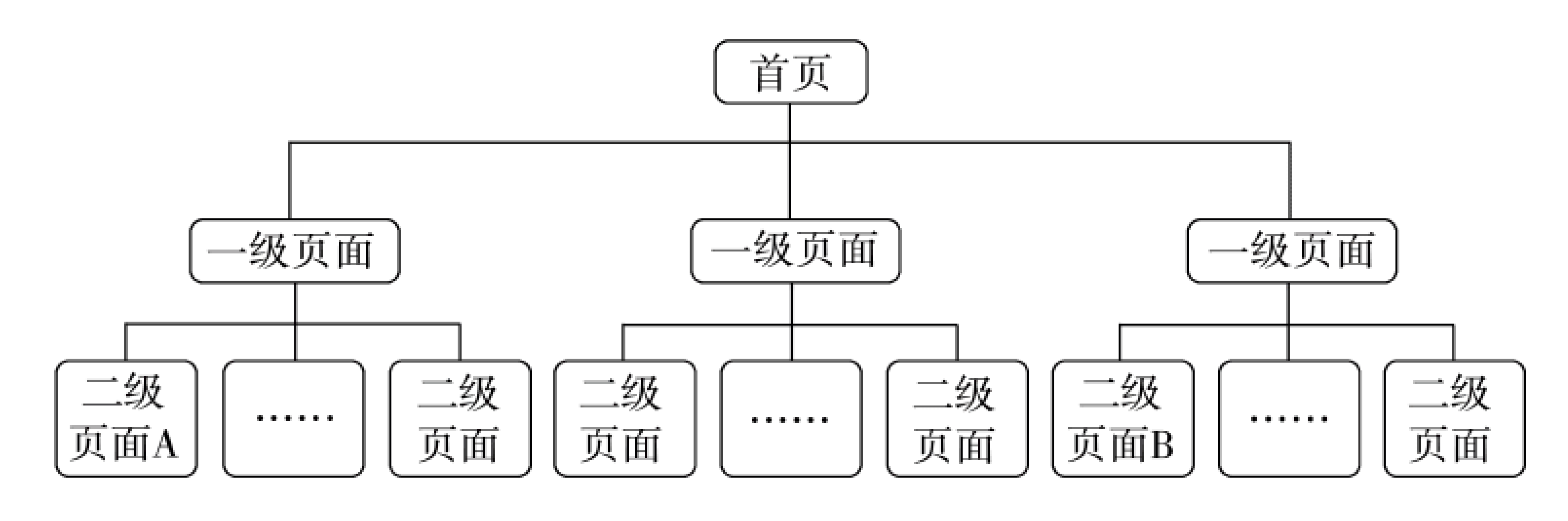


图1-3 树状链接结构图

(2) 星状链接结构(一对多)

每个页面相互之间都有链接，这种链接结构的优点是浏览方便，随时可以到达自己喜欢的页面；缺点是链接太多，容易使访问者迷路，搞不清自己处于网站的什么位置、已经看了多少内容等。星状链接结构简单示意如图1-4所示。

这两种链接结构是最基本的形式。在实际的网站设计中，总是将这两种结构混合起来使用。网站开发者希望浏览者既可以方便快速地到达想看的页面，又可以清晰地知道目前的位置，这就需要把这两种链接结构结合起来，形成一种更复杂的复合链接结构。

图1-4 星状链接结构图

(3) 复合链接结构

这种链接结构在主页和第一级、第二级页面之间采用树状结构链接，在二级以下的各页面之间采用星状结构链接。

1.6　确定网站的风格和创意

知识储备

网站的整体风格及创意设计是网站开发者们应该掌握但却很难掌握、很难应用的一种感性思维的体现，其难度在于没有固定的模式可以参照和模仿。对于一个网站主题，任何两个设计者都不可能创作出风格完全一样的网站。

网站的风格和创意不能光靠理性的设计来完成，它是表现于很多细微之处综合形成的一种整体形象，需要设计者具有广告设计、美学等方面的知识和经验，感兴趣的读者可以找一些广告设计、美学方面的书来阅读参考。

1.6.1　任务描述

确定“新闻类别网站”的风格和创意。

1.6.2　任务实现

考虑网站是新闻类网站，故确定其风格为守旧的、约定俗成的特点，如严肃、整齐、正统、规范；再考虑网站的新闻是“红色革命”题材方面的新闻，确定网站主要与“红色”和“革命”相关。因此，创意点可立足于红色、五角星、军装、镰刀和锄头等思路。

基于上述考虑，网站风格的设计将围绕新闻主题展开，遵循严肃、正统而又紧扣革命色彩的理念。在页面的整体设计上采用红色系配以灰色系，以突出红色的革命和灰色的革命武装服饰，配合使用矩形式的新闻显示区域。为了使矩形区域的显示不至于生硬，对矩形采用了圆角处理，并对列表矩形的标题部分采用了渐变的效果。整体的排版采用整齐划一的田式划分，又使整个页面归于严肃、正统。

为了给页面增加生机，使网站显得更加活泼，特设了一个滚动新闻板块，使整个页面不再死板。

1.6.3　相关知识拓展

1. 网站风格

网站风格是抽象的，是指网站的一种整体形象、整体风格。

网站风格包括站点的 CI、版面布局、浏览方式、交互性、文字、语气、内容价值、存在意义、站点荣誉等诸多因素。有的网站看上去平易近人，如网易；有的网站看上去生动活泼，比如迪斯尼；有的网站看上去专业严肃，如 IBM。网站不同的风格即体现为网站给浏览者留下的不同的综合感受。



网站风格必须是独特的，这是不同于其他网站的主要特色。可以通过色彩、技术或者交互方式等的表现，让浏览者能够明确分辨出这是我们的网站所特有的“气质”。

网站风格也是有人性的。通过网站的外表、内容、文字、交流，可以表现出一个站点的“个性特征”、“情绪特征”，网站是活泼易变的个性，还是执著热情的个性？是温文儒雅的个性，还是放任不羁的个性？

尽量使网站有特定的风格，不要仅仅是堆砌在一起的信息，要让网站从信息量大小、浏览速度快慢的理性评价中扩展开来。在网站中展现网站风格，好比给理性的人注入感情，使网站更具有亲和力和吸引力。

2. 网站创意

创意不是某些天才者的灵感，而是众人思考的结果。创意是传达信息的一种特别方式，可以通过将网页现有的要素重新组合的方式体现创意。平时注意资料的收集，资料丰富了，会更容易产生创意。最常用的创意思考方法是联想，下面列出作者从网上搜集的网站创意的 25 种联想方式：

①颠倒网站的某些对象；②缩小网站的某些对象；③换一下网站某些对象的颜色；④使对象更长一些；⑤使对象闪动；⑥把对象放进音乐里；⑦结合文字、音乐、图画；⑧使对象年轻化；⑨重复对象；⑩立体化对象；⑪参加竞赛；⑫参加打赌；⑬变更网站的一部分对象；⑭分裂对象；⑮罗曼蒂克化对象；⑯使网站速度加快；⑰增加网站香味；⑱使对象看起来流行；⑲使某些对象对称；⑳使网站展现出向儿童诉求的特征；㉑使相关对象价格更低；㉒给对象起个绰号；㉓把一些对象打包；㉔免费提供某些对象；㉕以上各项的延伸组合。

1.6.4 经验总结

网站不能没有风格，更不能没有内容，树立网站风格时，要确信网站风格是建立在有价值的内容之上。

要确定网站风格，一定要先考虑清楚网站希望树立的形象是什么。

将网站风格体现在网站的各个细节上，Logo、标准色彩、标准字体、宣传标语、统一的语气和人称、统一的图片处理效果、网站特有的符号或图标等。

不论如何创意，都必须清楚创意的目的是为了更好地宣传推广网站，是为提高网站知名度服务的。

1.7 确定网站的系统配置

1.7.1 任务描述

确定“新闻类别网站”的系统配置信息。



1.7.2　任务实现

（1）硬件环境

- 建议硬盘空间：1GB(包括搭建网站运行环境所需要的软件占用空间)。
- 建议内存：256MB(推荐 512MB 或者更高)。
- 建议 CPU：1.4GHz(推荐 2.0GHz 或者更高)。
- 网络环境：广域网或者局域网。

（2）软件环境

- 操作系统：Microsoft Windows 2000/NT/XP 或者更高版本。
- 数据库系统：SQL Server 2005。
- 其他支持软件：Tomcat 6.0.28＋JDK1.5。

（3）开发环境

- 开发软件：eclipse-jee-ganymede-win32。
- 服务器软件：apache-tomcat-6.0.18。
- JDK：jdk-6u10-windows-i586-p。

（4）其他的系统配置信息

系统的运行参数，如，每个页面显示新闻的条数、站点的搜索关键字、站点描述、作者等配置信息，这些信息将保存在网站数据库的“系统配置表”中，实现系统的自动配置。

1.7.3　相关知识拓展

网站配置是设置网站各管理对象的配置参数，并集中起来以便统一管理。网站配置参数包括以下配置参数：网站信息配置、网站参数配置、商店参数配置、用户参数配置、邮件参数配置、网站中的缩略图参数配置、IP 访问限定配置、手机短信配置、数据字典。在网站运营前，请根据需要详细配置好相关参数以保障网站的有效运营。

1.7.4　经验总结

一般来说，系统配置主要取决于系统的功能。在功能确定后，计算出系统的最低配置，并以此最低配置作为系统开发时的系统环境；而在正式部署时，也以此最低配置为准。但是为了使系统的功能达到更好，一般正式运行的系统配置都高于开发时的最低配置。

所以，在满足系统功能需求的基础上，要尽量降低开发时的系统配置。

1.8　搜集素材

1.8.1　任务描述

收集“新闻类别网站”需要的各种媒体类型的素材。



1.8.2 任务实现

尽可能多地从网络上或者其他的地方收集“新闻类别网站”需要的各种媒体类型的素材。

小贴士

收集时要注意与前述各节确定的网站风格、网站结构、网站主题、创意等相吻合，不要过于盲目。

1.8.3 相关知识拓展

网站设计需要相关的资料、素材以及丰富的内容才可以丰富网站的版面。个人网站可以整理个人的作品、照片、展示等资料。企业网站需要整理企业的文件、广告、产品、活动等相关资料。整理好资料后，需要对资料进行筛选和编辑。

可以使用以下方法来收集网站资料与素材。

(1) 图片

可以使用照相机拍摄相关图片，对已有的照片可以使用扫描仪输入到计算机。一些常见图片可以在网站上搜索或下载。

(2) 文档

收集和整理现有的文件、广告、电子表格等内容。对纸质文件需要输入到计算机中形成电子文档。文字类的资料需要进行整理和分析。

(3) 媒体内容

收集和整理现有的录音、视频等资料。这些资料可以作为网站的多媒体内容。

1.8.4 经验总结

明确了网站的主题以后，就可以围绕主题开始搜集材料了。常言道：“巧妇难为无米之炊”。要想让自己的网站有血有肉，能够吸引住用户，就要尽量多且广地搜集材料，搜集的材料越多，制作网站就越容易。材料既可以从图书、报纸、光盘、多媒体上得来，也可以从互联网上搜集，然后把搜集的材料去粗取精，去伪存真，作为自己制作网页的素材。

收集内容包括：和网站主题相关的文字、图片、资料；一些优秀的页面风格；一些喜欢的交互页面；开放的源代码等。

1.9 实　训

请读者思考并创建一个个人网站，按照本章的内容试着为这个网站做需求分析，并记录所有的分析资料和分析结果。

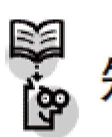


第2章　网站设计

在全面考虑好网站的栏目、链接结构和整体风格之后，就可以正式动手设计网页了。

俗话讲得好："良好的开端是成功的一半。"在网站设计上也是如此，首先要根据系统的需求确定网站的功能模块，其次根据主题、CI、风格、创意等整体风格确定网站的版面，再进行页面的划分以及页面层次的设计。页面设计中以首页的设计最为关键，其他页面的设计遵循风格一致的原则，在首页设计好后，根据功能的区别，依据首页设计方法进行。

2.1　设计网站的功能模块



知识储备

第1章已经确定了"新闻类别网站"的功能模块，这里的任务是考虑这些模块的实现方式。"新闻类别网站"的设计目标是尽可能地减少网站的后期维护成本，能很方便地发布企业的最新信息和其他各类信息，并可对这些信息进行删除、修改、添加等操作。

目前许多企业发布新闻的方式大多是采用静态的HTML页面，每次更新新闻信息时需要做一个页面，然后用FTP上传至服务器，再修改页面链接，使新增的新闻页面能正常显示。但这样的维护方式非常麻烦。建立"新闻类别网站"系统可以大大减轻网站更新维护的工作量，通过新闻信息的后台维护程序，网站管理员只需要在浏览器上录入新闻文字就可以快速实现新闻更新，从而大大加快了信息的传播速度，时时保持网站的活动力和影响力。

2.1.1　任务描述

设计"新闻类别网站"的功能模块。

2.1.2　任务实现

"新闻类别网站"采用B/S模式，网站的页面在浏览器中显示，网站的功能代码在服务器端运行，网站使用的新闻数据存放于数据库中。

(1)"新闻类别网站"前台功能设计

前台为浏览者提供浏览网站新闻的平台。这部分功能设计主要是网站前台版面的设计，包括网站栏目的划分、色彩的搭配、框架的设计、网站风格创意的体现等。这些内容具体



请参考本章后面的知识。

(2)"新闻类别网站"后台功能设计

- 用户管理模块：管理并维护网站新闻信息的用户身份，本系统中称此类用户为管理员。管理员身份存储于数据库中，只有登录成功的用户才能使用下述功能对网站的后台数据进行操作。
- 栏目模块：用于管理网站的栏目分类情况。可以由管理员在后台进行编辑，编辑后自动更新前台页面的显示结果。
- 专题模块：用于管理网站的专题分类情况。可以由管理员在后台进行编辑，编辑后自动更新前台页面的显示结果。
- 新闻模块：网站新闻可以动态增加、修改或删除。这些功能在网站的后台功能中实现，管理员是操作这部分功能的主要网站用户，浏览者可以在网站页面上查看到更新后的新闻。新闻分为热点新闻、图片新闻，不同类别的新闻将在网页的不同区域进行显示。管理员可以修改这些新闻属性的设置，决定该新闻是否出现在网站的热点新闻栏目还是图片新闻栏目中。
- 友情链接模块：作为网站的一个功能模块，主要在网站的页面上建立与其他网站的链接，形成友情交换链接。这部分内容不在数据库中进行管理，在维护时直接在网页上进行修改，然后上传修改后的页面。
- 系统日志管理模块：由服务器自动记录网站操作员的操作情况，这个功能只能供管理员在后台查看，不能修改，也不能删除。
- 系统配置管理：对网站整体的配置信息进行管理，如：Logo 的更换、版权信息、关键字、作者、描述、页面显示的新闻数等，只有管理员才有资格设置这些配置信息。

2.1.3 相关知识拓展

在第 1 章里定位了网站主题，确立了网站的 CI 形象、风格和创意等，收集了相关的素材，接下来是否该进入实质性的设计制作阶段呢？答案是：不能。经验告诉我们，建立一个网站好比写一篇文章，首先要拟好提纲，文章才能主题明确，层次清晰；也好比造一座高楼，首先要设计好框架图纸，才能使楼房结构合理。

初学者最容易犯的错误是确定网站题材后立刻开始制作。当辛苦地一页一页制作完毕后却发现：完成的网站存在结构不清晰、目录庞杂、内容是东一块西一块地拼凑的结果。不但使浏览者在网站上看得糊涂，连网站的后期扩充和维护工作也变得相当困难，最不幸的是网站或许就此半途而废，成为一个失败的例子。

因此，在动手制作网页前，一定要设计好网站的功能模块，考虑好栏目划分与未来的设计细节。

2.1.4 经验总结

1. 网站栏目安排的注意事项

栏目的实质是一个网站的大纲索引，能够将网站的主体明确显示出来。在制定网站栏目的时候，要仔细考虑，合理安排。



(1) 紧扣主题。将网站主题按一定的方法分类，并将这些分类后的主题作为网站的主栏目。

(2) 设置最近更新、网站指南栏目。如果网站首页版面中没有显示“最近更新”的位置，就有必要单独设立一个“最近更新”的栏目，方便常来的访客，让网站首页更具人性化。如果首页内容庞大(超过 15MB)、层次较多，但又没有设置站内搜索引擎时，建议添加“本站指南”栏目，方便初访者快速找到他们想要的内容。

(3) 设置双向交流栏目。可以在网站中加入如论坛、留言本、邮件列表等栏目，方便浏览者留下他们的信息，提高网站的亲和力。

(4) 设置下载或常见问题问答栏目。在网站首页上设置一个资料下载栏目，让好的资料共享；对于浏览者的共性问题，可在网站设置常见问题问答栏目，这样可增加网站的信誉。

(5) 其他注意事项。网站的其他辅助内容如关于本站、版权信息等，建议不要放在主栏目里，以免影响网站主题的吸引力。

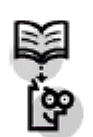
总之，要尽可能删除与网站主题无关的栏目，尽可能将网站最有价值的内容明显地列在首页栏目上，尽可能方便访问者的浏览和查询。

2. 版块设置的注意事项

版块比栏目的概念要大，网站内容复杂时，可以先划分版块，每个版块再分栏目。比如搜狐网站分新闻、军事、男人、女人、体育、彩票、旅游等版块，每个版块下面又各有自己的主栏目。但如果网站内容较少，只设主栏目(主菜单)就可以了，不必设置版块。

设置版块时要注意各版块既要保持相对的独立性，又要保持相互关联，并且各版块的内容要围绕网站主题，不要脱离主题。

2.2　设计网站的版面



知识储备

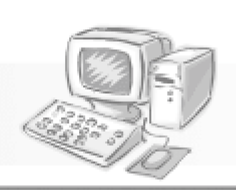
版面指的是在浏览器上看到的完整的一个页面(可以包含框架和层)，这个页面上的内容排列方式称为版面布局。版面设计是指在版面上有限的视觉空间中进行页面元素的排列组合，将理性的内容和思维个性化地表现出来。

网页作为一种版面，既有文字，又有图片。文字有大有小，还有标题和正文之分，图片也有大小，而且有横竖之别。网页上还有链接、动画等要素。这些要素需要同时展示给观众，按一定的次序进行合理编排和布局。

版面布局需要创意，但要比站点整体的创意要容易、有规律些。版面布局的步骤包括：

1. 草案

新建页面就像一张白纸，没有任何表格，框架和约定俗成的东西，设计者可以尽可能地发挥想象力，将想到的“景象”画上去(建议准备一张白纸和一支铅笔，如果用 Photoshop、画图工具等也可以)。这是页面的构造阶段，不必讲究绘画的细腻工整，不必考虑页面细节功能，只需以粗陋的线条勾画出网站版面的轮廓即可。尽可能多画几张比较比较，最后选定一个满意的作为继续创作的草案。



2. 粗略布局

在草案的基础上，继续将网站的功能模块安排到页面上。要安排的内容包括网站标志、主菜单、新闻、搜索、友情链接、广告条、邮件列表、计数器、版权信息等。安排时必须遵循突出重点、平衡谐调的原则，将网站标志、主菜单等最重要的模块放在最显眼、最突出的位置，然后再考虑次要模块的排放。

3. 定案

将粗略布局精细化、具体化。

2.2.1 任务描述

手工勾画“新闻类别网站”网站的版面。

2.2.2 任务实现

(1) 草案

“新闻类别网站”版面布局的草案如图 2-1 所示。

(2) 粗略布局

“新闻类别网站”版面的粗略布局如图 2-2 所示。

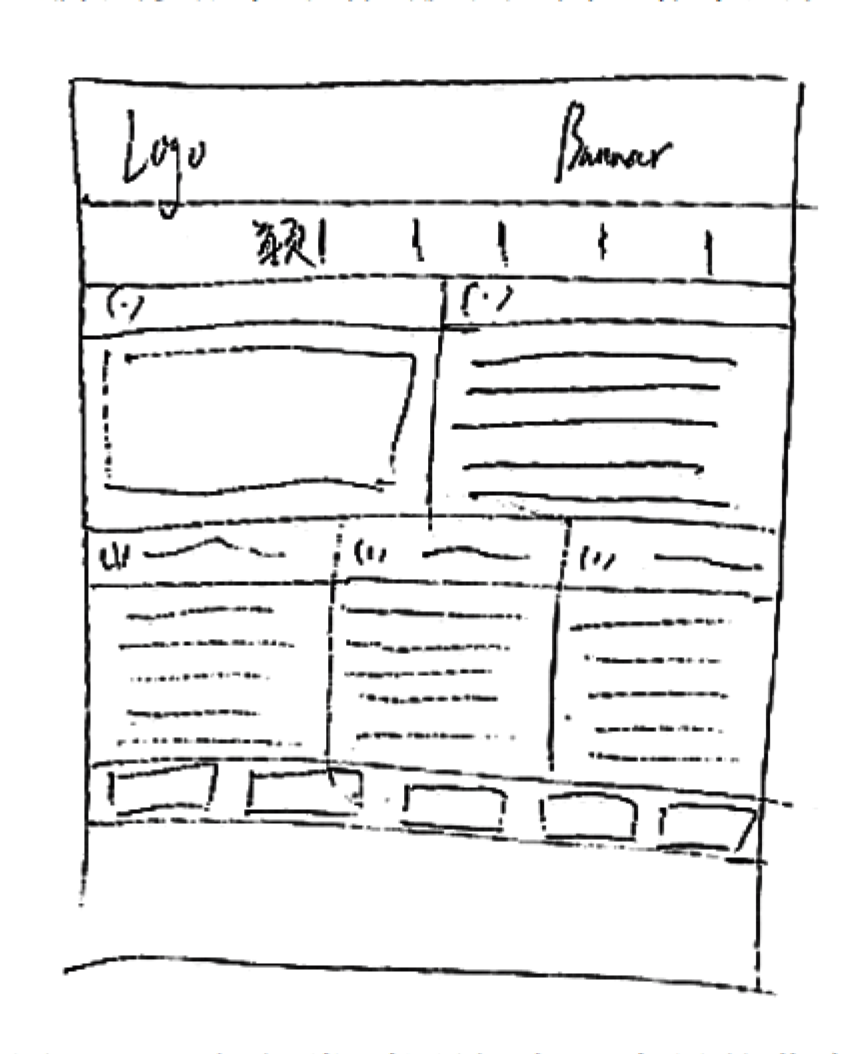


图 2-1 “新闻类别网站”版面布局的草案

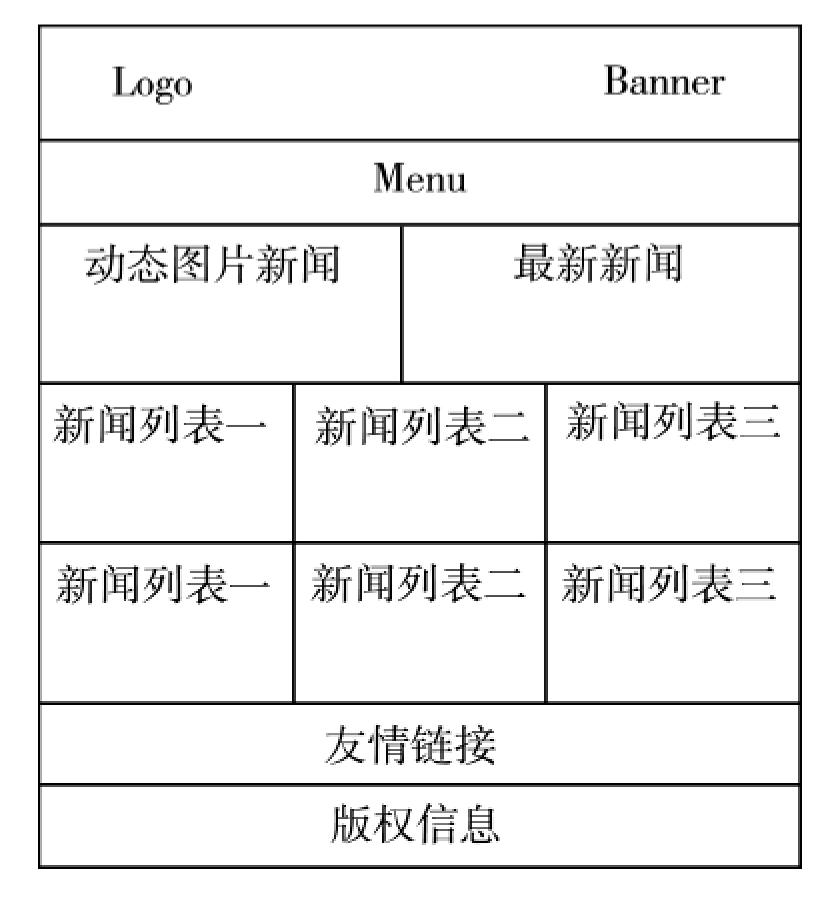


图 2-2 “新闻类别网站”版面的粗略布局

(3) 定案

将“新闻类别网站”版面的粗略布局精细化、具体化。这就要体现仁者见仁、智者见智的功力了，不同的设计者能创作出具有不同创意的版面布局。这里不再细述。

2.2.3 相关知识拓展

版面设计是网站设计的重要组成部分，表面上看，它是一种关于版式编排的学问，而实



际上，它不仅仅是一种技能，更是技术与艺术的高度统一。随着互联网的蓬勃发展，人们对设计的要求越来越高，高格调、高品位的设计作品，才能吸引住人们的眼球，而那些粗制滥造、毫无内容的作品，只能飘移在互联网的海洋中无人问津。

1. 几种常见的版面布局

版面布局是指以最适合浏览查看的方式将图片和文字排放在网页上的不同位置。经常用到的版面布局形式有：

（1）“T”型结构布局。“T”型结构是指页面顶部为横条网站标志和广告条，形成“T”的上面一横；页面下方，左面显示主菜单，右面显示内容。因为菜单条背景较深，整体效果类似英文字母“T”，所以称这种布局为“T”型布局。这是网页设计中用得最广泛的一种布局方式。“T”型布局的优点是页面结构清晰，主次分明；缺点是规矩呆板。

（2）“口”型布局。“口”型布局是指页面上下各有一个广告条，左面是主菜单，右面放友情链接等，中间是主要内容（有的“口”型布局是将四边空出，只用中间的窗口进行设计）。这种布局的优点是充分利用版面，信息量大。缺点是页面拥挤，不够灵活。

（3）“三”型布局。“三”型布局多用于国外站点，国内站点用得不多。特点是页面上横向两条色块，将页面整体分割为四部分，色块中大多放广告条。

（4）对称对比布局。对称对比布局采取左右或者上下对称的布局，一半深色，一半浅色，一般多见于设计型站点。优点是视觉冲击力强，缺点是将两部分有机地结合比较困难。

（5）POP 布局。POP 引自广告术语，POP 布局是指页面布局像一张宣传海报，以一张精美图片作为页面的设计中心。常用于时尚类站点。其优点是页面漂亮吸引人。缺点就是网页打开速度慢。

2. 版面设计的原则

（1）版面正常平衡。这种版面亦称“匀称”版面，整个版面形成左右、上下对照，秩序井然，体现安定诚实、信赖的效果。

（2）版面异常平衡。这种版面属非对称形式，但也讲究不均整的平衡和韵律，体现强调性、不安性、高注目性的效果。

（3）版面对比。在版面上利用色彩、色调等技巧来体现对比，也可利用内容体现古与今、新与旧、贫与富等对比。

（4）版面凝视。可以利用页面中人物视线，吸引浏览者仿照跟随的心理，达到注视页面的效果，一般多用明星凝视状。

（5）版面空白。版面空白能体现出网站突出卓越、有网页品位的优越感，能充分表现出网页的格调。

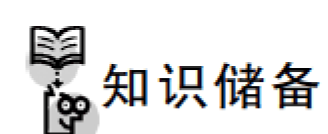
（6）版面尽量用图片解说。用图片表现不能用语言说服，或用语言无法表达的情感。图片能够展示、传达给浏览者更多的心理因素影响。

2.2.4　经验总结

在网站的版面布局上要注意加强各对象展现的视觉效果，加强文案的可视度和可读性，整体网站形成一个统一的感觉，要不时地更新网站，保持网站新鲜的内容和独特的个性。



2.3 设计网站的页面框架



知识储备

Fireworks 是网页三剑客(Dreamweaver、Fireworks、Flash)之一,原为 Macromedia 公司所有。在 Macromedia 公司被 Adobe 公司兼并之后,Adobe 公司又进一步发展了此软件。Fireworks 现在的最新版本是 Adobe Fireworks CS4。

Fireworks 是一个强大的网页图形设计工具,可以使用它创建和编辑位图、矢量图形,还可以非常轻松地做出各种网页设计中常见的效果,比如翻转图像、下拉菜单等,设计完成以后,若设计结果要在网页中使用,可以将它输出为 HTML 文件,还能输出为可以在 Photoshop、Illustrator 和 Flash 等软件中编辑的格式,从而实现与网页的无缝链接,为用户一体化的网络设计方案提供支持。

2.3.1 任务描述

利用 Adobe Fireworks CS3 进行"新闻类别网站"页面框架的设计,最终设计效果如图 2-3所示。

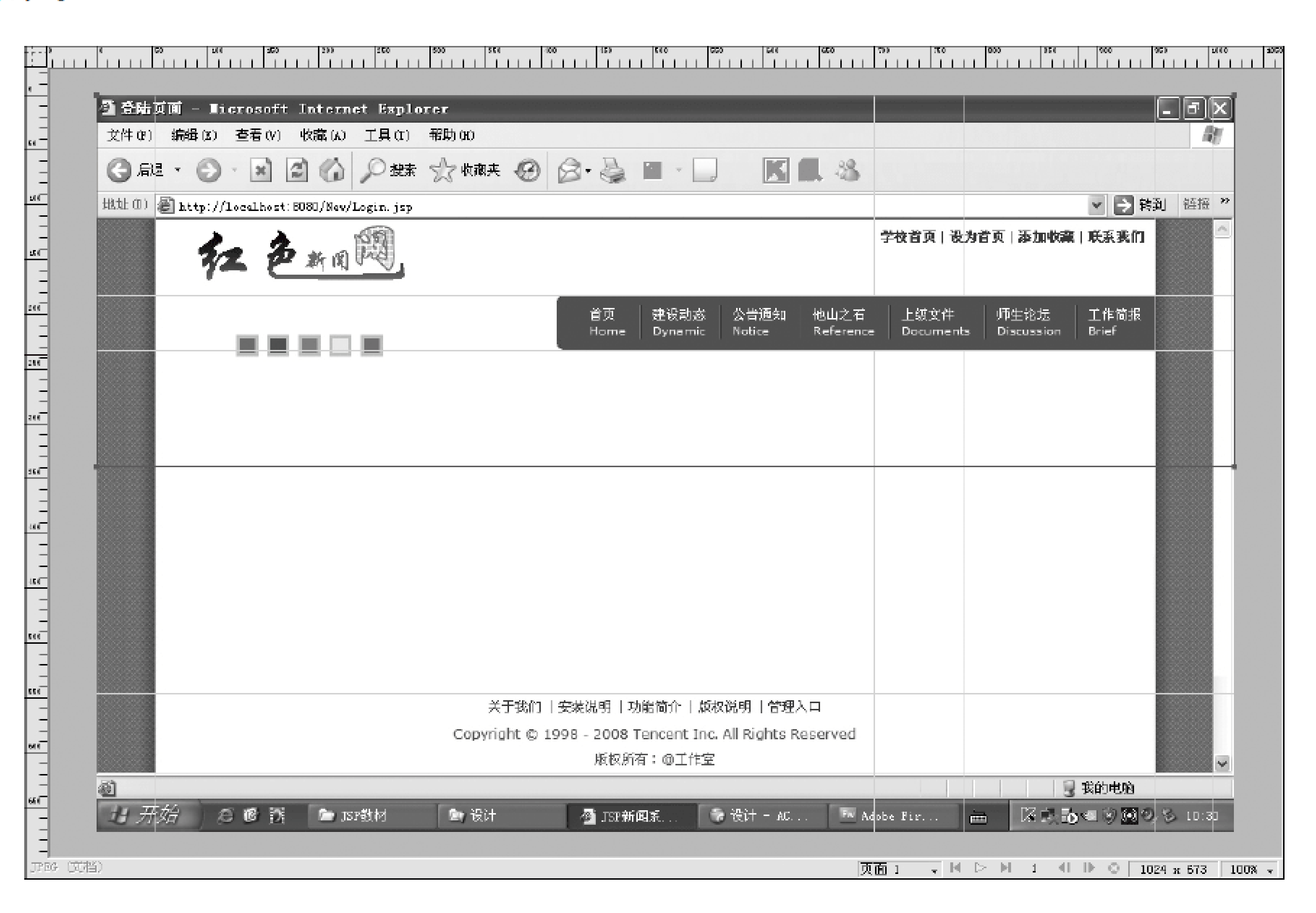


图 2-3 "新闻类别网站"的最终设计效果

2.3.2 任务实现

(1) 新建设计页面

① 打开浏览器,最大化后,截取 1024×768 像素的分辨率下浏览器满屏的视图作为页





面设计的基本框架。

② 打开 Adobe Firework CS3，新建一页面，将截取的图像置入 Firework 中，并删除浏览器主区域中的内容，形成一个只有浏览器框架的图像。

③ 用辅助线和标尺把大概范围划分出来，注意这里的划分，正好是在 770 像素和 1000 像素之内的地方，因为"新闻类别网站"是在 1024×768 像素的分辨率下运行的。考虑到可能存在使用 800×600 像素分辨率的浏览者，所以把页面上主要的内容和菜单都放置在 770 像素以内，770～1000 像素的地方放一些辅助的东西。这一点尤其是在做一些门户网站设计的时候更要重视。

在此，设计页面的宽度为 900 像素，并采用黄金分割进行划分，这样就将 900 像素的宽度划分为 630 像素和 270 像素，使左面的主要区域 630 像素也小于 770 像素，正好符合不同主流分辨率显示器的现实，然后再依据标尺建立辅助线。"新闻类别网站"的基本框架设计如图 2-4 所示。

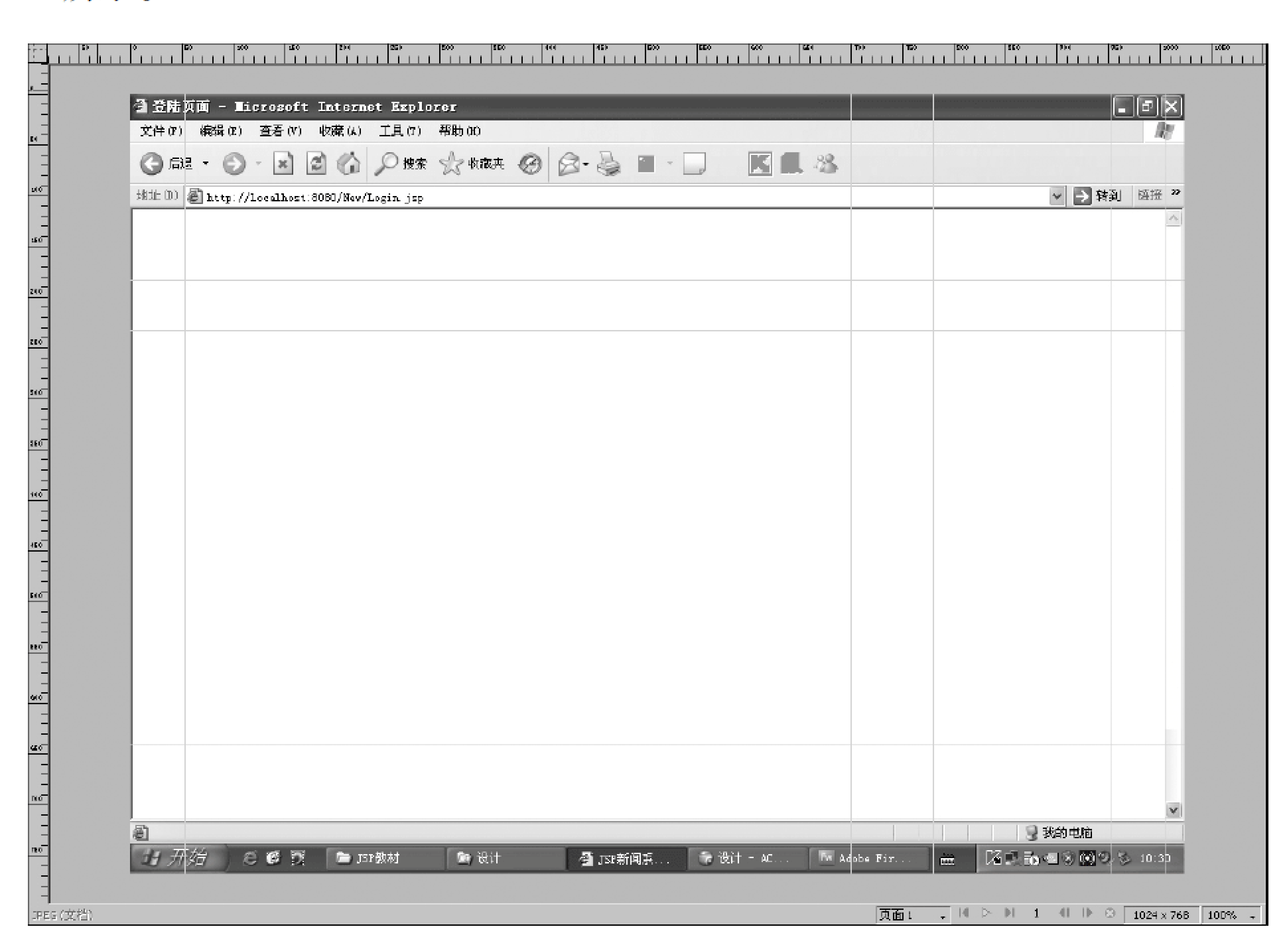


图 2-4　"新闻类别网站"的基本框架设计

小贴士

制作网页时，其页面大小取决于主流显示器的分辨率设置。

1. 页面高度

一般来说，页面的高度是无法进行精确限制的，因为每个人的浏览器上放置的工具栏不同，工具栏越多，网页显示的高度就越少，有些浏览器界面上仅工具栏、插件就占了半个屏幕，所以高度没有确切值。具体可依据内容的多少和人们的浏览习惯将页面的高度控制在一屏到三屏之间。

当采用单屏页面时，要特别注意页面在不同分辨率显示器下的显示效果，应采取适当的



措施，如百度的首页采用的就是单屏页面，使它在任何分辨率的显示器下都是居中的，不论水平还是垂直。

2. 页面宽度

① 在 IE 6.0 下，宽度为显示器的分辨率减 21 像素，比如 1024 像素的宽度减 21 像素就变成 1003 像素。

② 在 Firefox 下，宽度为显示器的分辨率减 19 像素，比如 1024 像素的宽度减 19 像素就变成 1005 像素。

③ 在 Opear 下，宽度为显示器的分辨率减 23 像素，比如 1024 像素的宽度减 23 像素就变成 1001 像素。

所以如果显示器是 1024 像素的分辨率，不如将网页设成 1000 像素；如果是 800 像素的分辨率，一般都设成 770 像素。

以上内容需要明白并且牢记，不然很可能做出来的网页不符合浏览器要求。还可以把页面高、宽度设定得再稍微小一点，以防有些浏览器加了插件或者其他的工具显示后，屏幕宽度会有变化，所以若是 800 像素的分辨率，一般可设定为 760 像素左右；若是 1024 像素的分辨率，可设定为 990 像素左右。

需要注意的是，在 IE 6.0(或更低)的浏览器上查看网页时，无论页面多高，都会有右侧的滚动条框，而 Firefox 或 Opear 在内容少于浏览器高度时不显示右侧滚动条框。

(2) 设计网站页面的背景

依据网站页面的整体风格和创意及相关的设计原则，对网站的整体背景进行设计，并考虑整个站点的背景使用，在此要从整个站点的角度来考虑，要尽量设计一种既简单又便于在网页中实现的背景，即从背景大小、形状以及展开方式上来考虑。下面进行具体设计。

① 在图 2-4 建立的文件框架下面新建一个大矩形，作为背景的实现层，此层主要用建立加有“交叉并行线”纹理的背景效果，具体属性设置如图 2-5 所示。



图 2-5　实现背景层的属性设置

② 在背景层上面的框架下，新建一个矩形作为主区域层，此层主要采用发光的滤镜效果为背景层添加立体效果，具体属性设置如图 2-6 所示。



图 2-6　“新闻类别网站”的背景设置



③ 至此，得到如图 2-7 所示的“新闻类别网站”的背景设计效果图。

图 2-7　“新闻类别网站”的背景设计效果

（3）设计网站的头部、导航和尾部

下面分别对网站的头部、导航和尾部进行设计。要注意整个页面的比重，避免出现头重脚轻的局面。Logo 的设计要符合网站的主题和目标，而且要简明扼要、紧扣宗旨；导航要尽量直观、简洁、明显，最主要的是使用要便捷；尾部内容对网站做补充说明，设置版权等内容。下面进行具体设计。

① 利用辅助线勾勒出头部、导航和尾部的大致区域，主要依据导航醒目、内容优先、保持比例、避免失衡的原则。

② 在页面顶部制作 Logo，输入相应的文字，并采用渐变效果，而后将文字选中，右击并选择【转换为路径】命令，再对转换为路径的文字进行自由变形处理，形成最终的效果。

③ 制作主导航。主导航的制作要醒目、明快而简洁，并且始终要保持可见，即：在任何地方都要能看见主导航，一般情况下，导航的制作涉及超链接，利用超链接悬停状态的不同属性可以得到很好的效果，如图 2-8 所示。图中的中间隔线的效果是采用了两条细线并排来实现的，而颜色相对于底色为一深一浅。



图 2-8　“新闻类别网站”的主导航设计

④ 在页面的底部输入尾部内容，如：“关于我们”、“版权说明”等相关的补充说明性内容，一般居中放置，并用细线与内容部分隔开，效果如图 2-9 所示。

将上述部分进行合理的布局，便得到了“新闻类别网站”页面框架的效果图，如图 2-3 所示。



图 2-9 "新闻类别网站"的尾部设计

2.3.3 相关知识拓展

1. 色彩的基本知识

众所周知，大自然的颜色是因为光的折射而产生的。在计算机中红、绿、蓝是基本色，其他的色彩都可以用这三种色彩调和而成。HTML 标记语言中的色彩表达也是用这三种颜色的数值来表示。比如，红色是 red，十进制表示为(255，0，0)，十六进制表示为(FF0000)；白色是 white，十进制表示为(255，255，255)，十六进制表示为(FFFFFF)，HTML 代码中的"bgColor=＃FFFFFF"就是指背景色为白色。任何色彩都有饱和度和透明度的属性，属性的变化会产生不同的色相，所以至少可以调出几百万种色彩。

颜色可以分为非彩色和彩色两类。非彩色是指黑、白、灰色，黑、白是最基本和最简单的搭配，白字黑底、黑底白字都非常清晰明了。灰色是万能色，可以和任何色彩搭配，也可以帮助两种对立的色彩和谐过渡。彩色是指除了非彩色以外的所有色彩，如红、绿、蓝、黄等。但彩色的记忆效果是黑白色的 3.5 倍，即彩色页面较完全黑白页面更加吸引人。一般建议网站中主要内容文字用非彩色(黑色)，边框、背景、图片用彩色。这样页面整体不单调，浏览主要内容时也不会眼花。

2. 色彩与心理

不同的颜色会给浏览者带来不同的心理感受。

- 红色：一种激奋的色彩，能产生刺激效果，使人产生冲动、愤怒、热情、活力的感觉。
- 绿色：介于冷暖两种色彩的中间，能表现出和睦、宁静、健康、安全的感觉。若与金黄、淡白搭配，可以产生优雅、舒适的气氛。
- 橙色：一种激奋的色彩，能产生轻快、欢欣、热烈、温馨、时尚的效果。
- 黄色：能表示出快乐、希望、智慧和轻快的个性，它的明亮度最高。
- 蓝色：一种最具凉爽、清新、专业的色彩。若与白色混合，能体现柔顺、淡雅、浪漫的气氛。
- 白色：能产生洁白、明快、纯真、清洁的感受。
- 黑色：能产生深沉、神秘、寂静、悲哀、压抑的感受。
- 灰色：能产生中庸、平凡、温和、谦让、中立和高雅的感觉。

可以在每种色彩的饱和度和透明度上略微变化，以产生更多不同的感觉。比如黄绿色有青春、旺盛的视觉意境，而蓝绿色则显得幽静、深邃。

3. 网页色彩掌握的过程

一般在刚开始学习网页制作时，由于技术和知识的缺乏，只能制作出简单的网页，色彩



比较单一，这时的用色可以称为“单色”；有了一定的网页制作基础和材料后，又会将收集到的认为是最好的图片、最满意的色彩堆砌在网页上，造成网站色彩杂乱，没有个性和风格，这时的用色可以称为“五彩缤纷”；随着网页制作经验的积累，设计者会重新定位自己的网站，选择适合自己的色彩，使网站设计走向成功，这时的用色可以称为“标准色”；当设计理念和制作技术达到顶峰时，又会返璞归真，用单一色彩甚至非彩色就可以设计出简洁精美的站点，这时的用色又称为“单色”。

所以说，网页的色彩搭配是个人经验、设计理念、制作技术、创造性思维的综合表现，不是一日之功，初学者可多看网站、多研究、多总结，逐步提高对色彩的感知能力。

2.3.4　经验总结

虽然网站用色需要经验，但也不是没有任何技巧可言。下面推荐几种常见的配色方案以供参考。

(1) 选用一种色彩作为网站主色调。即先选定一种色彩，然后调整该色彩的透明度或饱和度，产生新的色彩，并将这些色彩都用于网站。这样的网站看起来既不特别的单调，而且色彩也统一，有层次感。

(2) 选用对比色作为网站主色调。对比色是两类拥有完全不同个性的颜色，如红和绿、蓝和橙、黑和白、紫与黄等。若有意将对比色搭配在一起，就要注意对比色间的比例变化，选择一种颜色为主色而另一种颜色为辅色，很有点睛的效果。若自己把握不好对比色，可以从Photoshop里得到，但切记，不要任意地使用两种不同的颜色作为网站主色调。使用两种合适的色彩会使整个页面色彩丰富但不花哨。

(3) 选用一个色系的色彩作为网站主色调。将同色系的颜色搭配在一起决不会出错，如“粉红＋大红”、“艳红＋桃红”、“玫红＋草莓红”等，这类同色系间的变化搭配，可制造出同色系色彩的层次感，又不会显得单调乏味，是比较简单易行的网站配色方法。

(4) 选用黑色和另一种彩色作为网站主色调。黑色是个百搭百配的色彩，无论与什么色彩放在一起，都会别有一番风情。

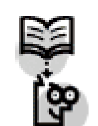
上面说的都是合理的搭配建议，下面再介绍一些不合理的搭配方案。

(1) 忌讳使用颜色对比过于强烈的搭配色作网站主色。

(2) 网站配色要简洁，不要试图将所有颜色都用到，主色要尽量控制在三种色彩之内。

(3) 网页背景和前景的对比要明显，尽量少用花纹复杂的背景图片，避免喧宾夺主，以便突出显示文字内容。

2.4　设计网站的首页



知识储备

首页，是整个网站内容的总目录，网站的版块划分均需要在此页面上展示，浏览者通过首页即可了解该网站的主要内容。设计首页的一般步骤如下。



1. 确定网站的功能模块

网站的功能模块是指需要在网站上展现的主要内容和功能。一般的站点都需要这样一些模块：

网站名称(Logo)、广告条(Banner)、主菜单(Menu)、新闻(What's new)、搜索(Search)、友情链接(Links)、邮件列表(Maillist)、计数器(Count)、版权(Copyright)

为网站选择哪些模块、实现哪些功能、是否需要添加其他模块，都是网站设计需要首先完成的工作。

2. 设计网站的版面

在网站的功能模块确定后，就开始设计网站的版面。设计版面的最好方法是：找一张白纸、一支笔，将理想中的版面草图勾勒出来，形成大致的框架布局。

3. 进行具体的界面设计

有了版面草图之后，再利用图形制作软件进行详细的设计，将理想中的草图进行细致的处理，实现直观而详细的设计图。

进行首页设计时，可以多参考一些好的设计效果，借鉴他人的长处，将别人的设计优点恰当地用在自己的设计中，形成自己的设计结果，切忌照搬照抄。

4. 处理设计技术上的细节

制作的网页如何能在不同分辨率下保持不变形，如何能在不同的浏览器下看起来不至于太丑陋，如何设置页面字体和链接颜色等，这些页面设计上的细节也必须考虑。

2.4.1 任务描述

利用 Adobe Fireworks CS3 进行“新闻类别网站”的网站首页的设计，最终设计效果如图 2-10 所示。



图 2-10 “新闻类别网站”网站首页的设计图



2.4.2　任务实现

具体操作步骤如下：

（1）制作列表框架，效果如图 2-11 所示。



图 2-11　“新闻类别网站”列表框架设计图

① 新建一个矩形，打开【属性】面板在【矩形圆度】文本框中输入“25”，其他属性设置如图 2-12 所示。

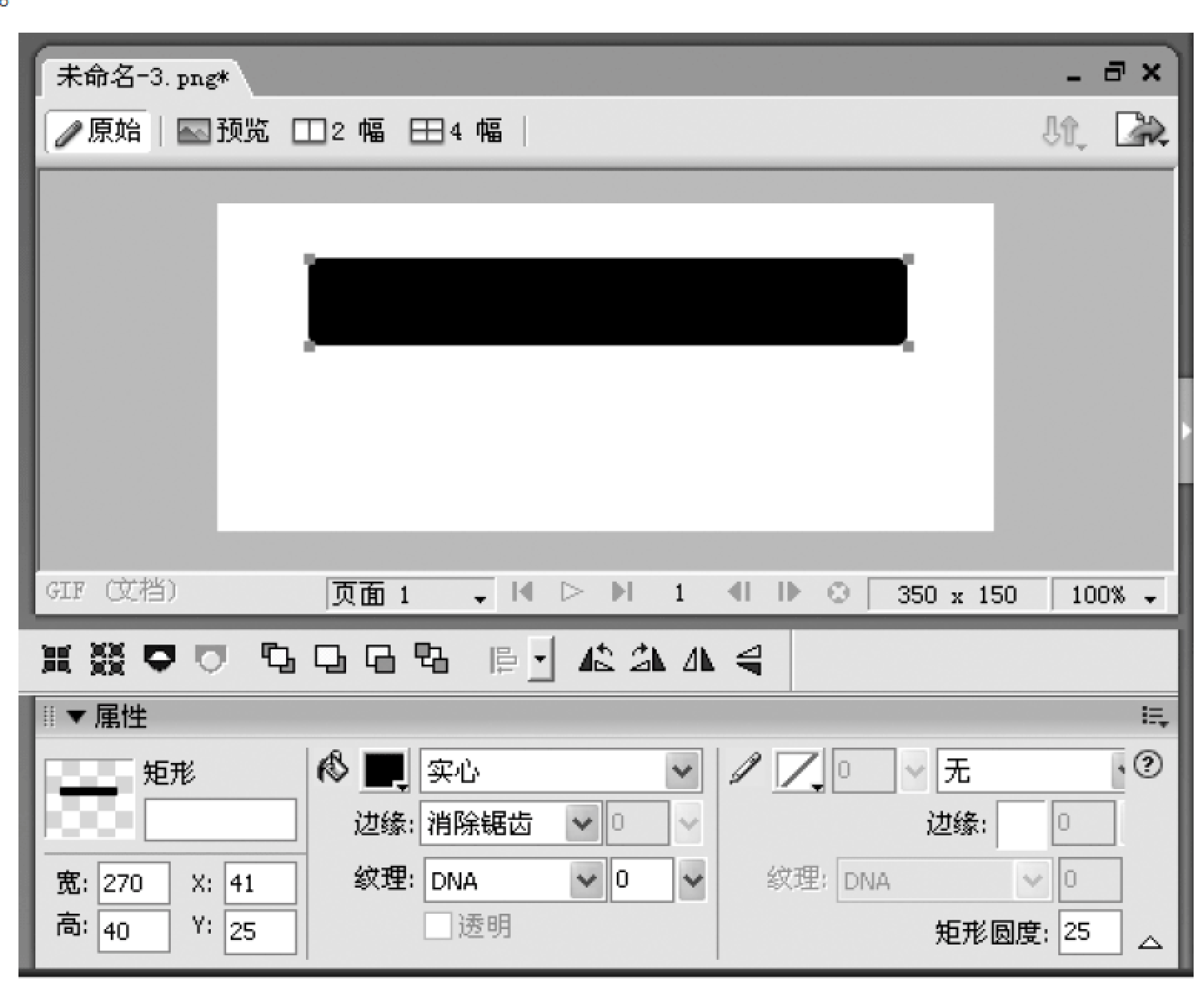


图 2-12　创建圆角矩形

② 在上面所建矩形的下方新建一个略大一些的矩形，可参照图 2-13 做具体的属性设置。

③ 同时选中上面所建的两个矩形，选择【修改】|【组合路径】|【打孔】命令，对其设置渐变效果，具体的属性设置如图 2-14 所示，这样做是为了获得理想效果。

④ 新建两个矩形，宽度和上面的一样，一个高度为 200、背景色为白色（#FFFFFF）；另一个高度为 2、背景色为红色（#CC0000）。添加标题栏上的图标和相应的文字，对上述元素进行排版，以实现如图 2-11 所示的效果，设置好后，选中所有对象进行组合，形成一



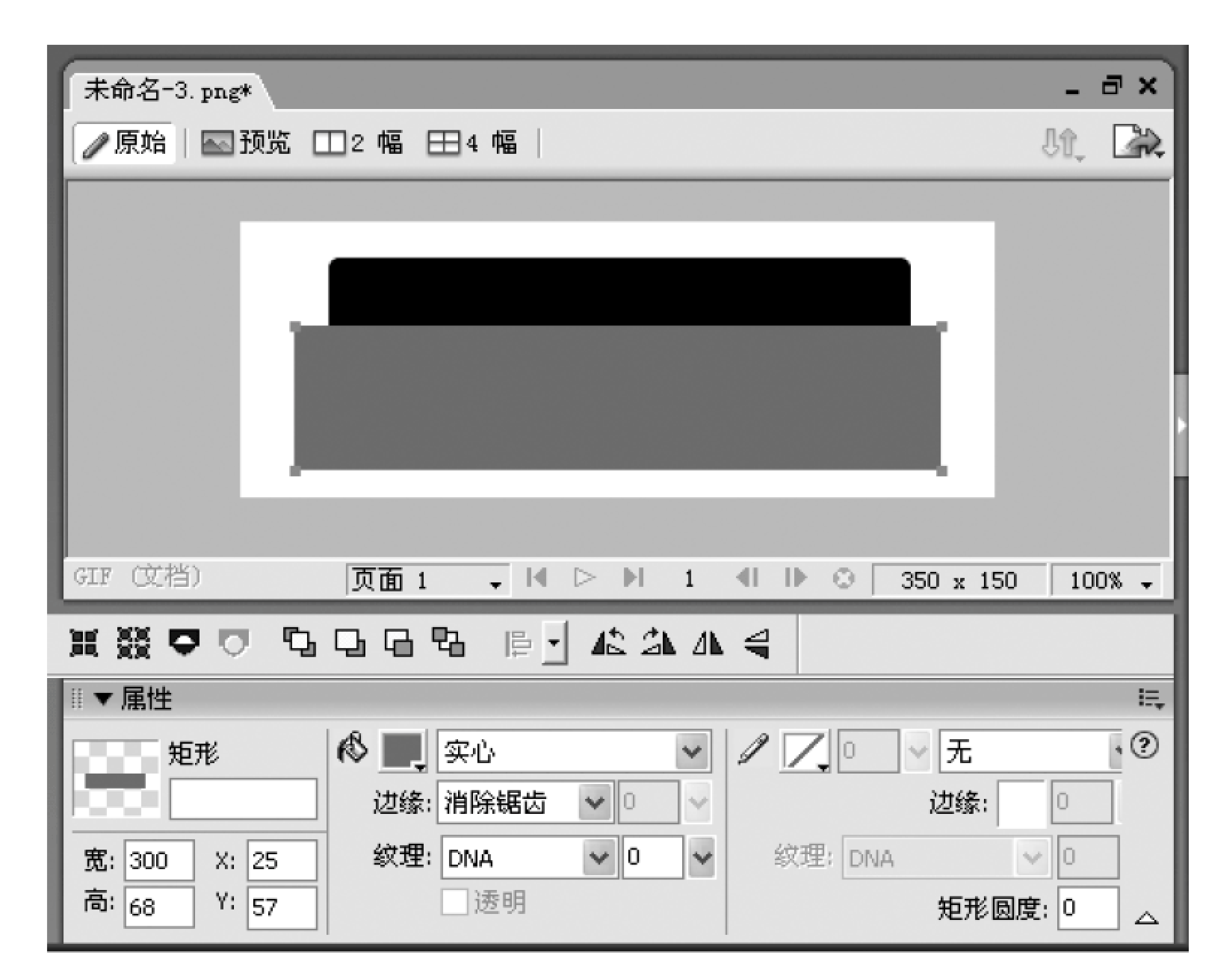


图 2-13　绘制第二个矩形

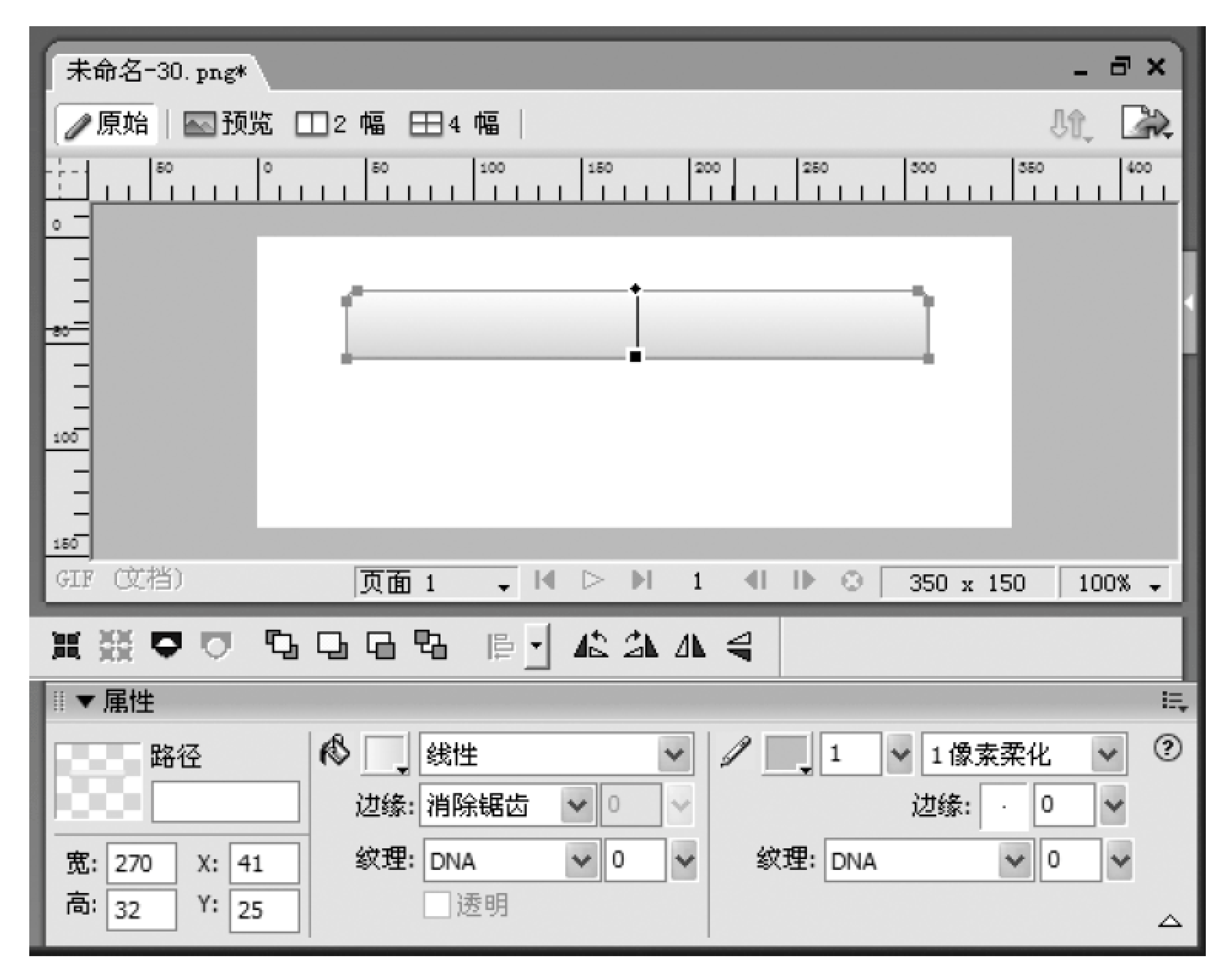


图 2-14　渐变设置

个整体。至此,列表框架制作完毕。

(2) 打开上面制作的"列表框架",导入一幅图片,调整其大小,实现如图 2-15 所示的"图片新闻"版块。

(3) 继续在"列表框架"中添加相关的文字,调整文字的属性,如图 2-16 所示。最后对文字进行排版,完成"最新新闻"版块的设计制作,如图 2-17 所示。





图 2-15　“图片新闻”版块的设计图



图 2-16　“最新新闻”版块字体属性的设置



图 2-17　“最新新闻”版块的设计图



(4) 打开图 2-4 制作的框架页面，进行网站首页的设计制作，这样既可以保持网站整体的风格一致，又可以准确定位相关模块的位置。

将图 2-15、图 2-17 制作好的“图片新闻”和“最新新闻”版块放入图 2-11 制作的框架页面中，并添加相对应的版块，最后排版、调整和拼接各相关版块，实现如图 2-10 所示的首页效果图。

2.4.3 相关知识拓展

1. 页面布局元素

在正式设计网站时，先考虑好网站的整体外观和页面结构，从视觉和技术两个角度确定组成整个页面的元素及元素之间的关系。页面布局要体现网站内容的分类情况，使内容能因为分类而显得简化，体现网站的逻辑性和易于理解性。当确定了页面布局元素之后，应该在整个网站上贯彻这些元素，使网站风格统一。

页面布局元素包括整体页面规划布局、页面边界内容与样式、字体、前景背景颜色、图像要求以及表格规格等。

- 框架元素：用于规划页面布局，每个框架里可以显示一个网页，框架的应用突破了原来浏览器只能显示一张网页的限制，应用了框架的网页能同时显示多个页面。
- 表格元素：常用于规划页面布局，准确定位页面中各对象的位置。
- 导航元素：用于充当网站内容的向导，帮助浏览者在网站内定位和浏览网页。
- 段落元素：和 Word 中的段落功能一致，将页面上的文本进行分段。
- 图像元素：比文本更直观，更具吸引力的网页元素。
- 列表元素：列表在网页中占有极其重要的地位，HTML 规定了多种列表元素，可根据文本内容的特征选择不同的列表进行展示。
- 颜色元素：提升网站品位的主要元素。
- 边距元素：控制网页所显示的内容距浏览器窗口边界的距离，类似于 Word 文档的上下左右边界。
- 边界元素：为 HTML 表和框架提供可见的边界线。
- 空白元素：空白元素用来分开页面上的各元素。一个没有空白区的页面，给浏览者的感觉是页面很拥挤，阅读时容易心理紧张，导致浏览者对这样的页面产生一种厌烦抵触的感觉。

2. 再论“空白”元素

初学网页设计的人往往会忽视“空白”元素，“空白”元素实际上与其他页面布局元素紧密关联，甚至是其他元素的一部分，比如行间距。空白在网页设计中非常重要，它使网页看起来更简洁、更明快，阅读起来更舒畅。因此“空白”是网页设计中必不可少的元素，在此再多费笔墨谈谈。

前面曾提过，在网页上不能堆积无用的内容。当浏览者快速扫描页面时，扫描的内容越多，错过的信息也越多。所以不要试图在一个页面上提供网站所有的信息。有一些商业网



站的主页提供特别大量的信息，反而会使浏览者忽视大部分内容。每一张网页的信息包含量应该是比同样印刷版本的信息量约少 50%。

网页上的信息可以通过分类的方法进行整合，避免让浏览者从一堆信息中艰难挑选所需要的信息，要让他们自己选择是否进入更深的链接获得信息。也不要只为了让页面变短而随意分割页面，除非在内容上是逻辑断点处。每一个页面都应该是相对独立的、完整的。只有照顾了整个页面空间的分配，空白才能表现出一定的活力，利用空白可以使页面布局生动活泼、松紧有度，使浏览者浏览页面时感觉到舒适、轻松和简洁。

网页空白可以使用表格和透明 GIF 图像来实现。使用表格定位页面内容时，可以设置不填充内容的空白单元格来产生页面空白，达到页面布局松紧有度的效果。

3. 页面布局设计方法

完成页面布局设计的方法主要有两种：纸面布局法和软件布局法。

（1）纸面布局法。在设计页面布局时，设计人员在纸面上画出页面布局的草图，将设想落实到纸面上，观看设计效果。人们经常会在瞬间产生一个好的灵感和想法，但这种想法若不及时记录下来，很快又会从头脑中消失。因此当有了好的灵感和想法时，应该立即用纸笔记录下来，以备以后用于页面设计中。

不要直接在网页设计工具中边设计网页边修改页面布局，想到哪里，设计到哪里，这样会浪费很多时间和精力。在开始制作网页前，应该首先在纸上画出页面布局草图，一旦确定就不要轻易修改，并在整个网站的页面设计中要自始至终贯彻确定好的页面布局。

（2）软件布局法。另外一种页面布局方法是使用图形工具软件设计页面布局草图，可以使用比较熟悉的图形软件如 Photoshop、Fireworks 等。使用这些图形工具可以方便地设计页面布局和颜色，比纸面布局法更能查看页面布局的整体效果和真实效果，使用图形工具和层更容易修改和查看各种无法用纸张实现的效果。

2.4.4　经验总结

首页的设计是一个网站成功与否的关键。人们往往看到网站第一页就对站点产生一个整体的感觉。能不能促使浏览者继续点击进入，能否吸引浏览者多花时间驻留在站点上，全凭首页设计的“功力”了。所以，首页的设计和制作是绝对要重视和花心思的。一般首页设计和制作要占整个网站制作时间的 40%。记住，应多花些时间在整体设计、框架制作和首页设计上，避免网站全部做好以后再修改，那将是最浪费精力、浪费时间、浪费金钱的事。

2.5　网站其他页面的设计

2.5.1　任务描述

利用 Adobe Fireworks CS3 进行“新闻类别网站”其他功能页面的设计。



2.5.2 任务实现

对于新闻类网站涉及三类重要的页面，即首页、列表页和内容页，由这三个页面的相互链接就构成了整个完整的新闻类系统。首页是显示所有栏目新闻的一个窗口，是整个站点的入口；列表页是针对某一栏目或专题而言的，是用于将该栏目或专题下的新闻以列表的形式显示出来的页面；内容页是用于显示新闻具体内容的页面，是最重要的页面。具体操作步骤如下：

(1) 对列表页进行设计，效果如图 2-18 所示。



图 2-18 “新闻类别网站”列表页设计图

先来划分这个列表页面，除了基本的框架外，其中的内容排列采用的是和“最新新闻”模块相同的形式，只是页面的具体内容和列表项目的数量不同，另外还增加了一个分页导航。

在此，还是采用老办法，先打开框架页面，在框架页面的基础上将图 2-17 做好的“最新新闻”列表框架的结构进行调整，制作出所需的模块样式，再将相应的内容添加进来进行页面排版，从而实现如图 2-18 所示的列表页效果图。

(2) 对内容页进行设计，效果如图 2-19 所示。

同上一步骤，先来划分内容页面，除了基本的框架外，其中“专题新闻”和“热点新闻”模块与列表页面中采用相同的结构，只是内容部分有所不同。

在制作上还是采用老办法，先打开框架页面，在框架页面的基础上将图 2-18 做好的“专题新闻”和“热点新闻”模块复制过来，重点在于实现内容部分的设计，从而实现如图 2-19 所示的效果。

内容页部分的模块也可进一步划分，并采用统一的结构，为套用程序做好准备，具体分为五块：标题、相关信息、导读、内容和前后导航。在实际的显示中，这种结构是不变的，只是具体的内容发生了变化。对这五个部分分别进行显示样式的设计，从而达到和谐一致的效果，如图 2-20 所示。



图 2-19　“新闻类别网站”内容页的设计图



图 2-20　设置显示样式

通过对各个版块的调整和整体的排版，从而实现如图 2-19 所示的内容页效果图。

2.5.3　相关知识拓展

1. 页面布局技术

常用的页面布局方法有：层叠样式表页面布局、表格页面布局、框架页面布局、透明 GIF 页面布局和 Flash 页面布局。



(1) 层叠样式表页面布局

在 HTML 4.0 标准中,层叠样式表(CSS)能完全精确地定位文本和图片。现在 CSS 已经成为一个比较流行的页面布局方法,可以实现过去无法实现的网页布局。越来越多的浏览器开始支持 CSS 技术,但不同的浏览器对 CSS 相关属性的解释可能略有不同,制作网页时应该注意。

(2) 表格页面布局

表格布局现在几乎已成为一个页面布局的标准,它的优势在于表格能对不同对象加以处理,而又不用担心不同对象之间的影响。表格的每一个单元格可以放置不同文本和图片,而且非常易于对齐。表格边界设置为 0,浏览时就看不到表格边框,只能看到表格单元格内放置的内容。使用表格进行页面布局比用 CSS 还要方便。表格布局唯一的缺点是,当用了过多表格时,页面下载速度会受到影响。

(3) 框架页面布局

框架可以用于页面布局,框架页面以左上角对齐,每一个框架部分都可以独立控制,但表格单元格却不可以单独控制。从布局上考虑,框架结构不失为一个好的布局方法。框架技术也可以设置边框为 0,浏览时也看不到框架的边框。

(4) 透明 GIF 页面布局

透明 GIF 图像可以作为空白区域的占位符插入到网页中,它们是背景透明、尺寸可变的图像,因此,透过图像可以看到页面的背景,浏览者是看不到透明 GIF 图片的。可以在<IMG>标记中指定透明 GIF 图像的高、宽属性来创建需要的空白的尺寸。例如,使段落从左面缩进 20 个像素,则在段落的第一个字符前插入下面的标记:

```
<IMG SRC="transparent.gif" HEIGHT="2"WIDTH="20">
```

也可以在表格的单元格中放置透明的 GIF 图像用于控制行或者列的尺寸。

(5) Flash 页面布局

这种页面布局采用了现在非常流行的 Flash 动画用于规划页面的布局,Flash 有强大的功能,使页面所表达的信息更丰富,其视觉效果及听觉效果要比其他页面布局更加具有冲击力,是当今最流行的一种技术,但是显示文字信息内容比较少,另外还需要更快的机器和更高的网络带宽。

2. 页面尺寸

由于页面尺寸与显示器大小及分辨率有关系,因此网页不能超越显示器的范围,而且因为浏览器本身也占去屏幕不少空间,所以留给页面的空间就会变得更加小。一般分辨率在 800×600 像素的情况下,页面的显示尺寸为 780×428 像素;分辨率在 640×480 像素的情况下,页面的显示尺寸为 620×311 像素;分辨率在 1024×768 像素的情况下,页面的显示尺寸为 1007×600 像素。从以上数据可以看出,分辨率越高,页面尺寸越大。

浏览器的工具栏也是影响页面尺寸的主要原因。一般浏览器的工具栏都可以显示或者隐藏,因而造成页面显示的尺寸也是不一样的。

在网页设计过程中,向下拖动页面是唯一给网页增加更多内容(尺寸)的方法。但除非需要,否则不要使用让访问者拖动很长页面的方法来增加页面的尺寸,页面过长,访问者往

往会忽视页面下面的内容。

另外,要切记在网页设计过程中,一定不要让页面过宽,造成左右拖动的页面。

2.5.4　经验总结

整体网站拥有一致的设计风格是很重要的,其中网站首页的设计是重中之重,在首页的风格和样式确定后,其他页面的设计都要遵循这种风格,使网站整体看起来和谐统一。

2.6　实　　训

按照本章的讲解过程,利用 Adobe Fireworks CS3 或其他作图软件完成"新闻类别网站"的框架设计工作,并设计网站首页和各主要页面。



第3章　实现网站数据库

在对网站进行操作时，有许多数据都需要存储起来。比如，在网上注册邮箱时，需要填写用户注册信息，在填写完成后，单击【提交】按钮，用户所填写的数据就会传给服务器，并存储到服务器的磁盘上。再如，在用户登录邮箱时，需要先输入用户名和密码，服务器根据用户提交的这两个数据进行身份验证，这个验证过程需要从预先存储的用户注册信息里进行查询，若存在该用户，则验证通过，否则验证不通过。上述操作都需要用到数据库管理软件，此类软件能够实现数据的存储并在此基础上实现增加、删除、修改、查询等常用的数据操作（当然数据库软件的功能远不止这些，这些只是它的最基本的功能）。

数据库缩写为DB，即Database，用来存放网站所需的后台数据。目前常用的数据库是关系型数据库，“新闻类别网站”采用微软的SQL Server 2005作为数据库的管理软件。

网站建设在本阶段的工作任务包括：确定网站的数据结构、创建数据库、创建表、分离和附加数据库。

3.1　确定网站的数据结构

3.1.1　任务描述

对“新闻类别网站”进行数据分析，确定此网站的数据结构需求。

3.1.2　任务实现

具体操作步骤如下。

（1）收集信息

要确定“新闻类别网站”的数据结构，首先要充分理解网站需要完成的任务和需要实现的功能，网站功能的实现需要后台数据库的支持。也就是说要考虑在数据库中存储什么数据，才能支持网站功能的实现。

“新闻类别网站”的功能具体包括以下几点。

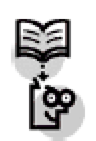
- 用户的注册和登录功能：这要求后台数据库存放用户的注册信息和在线状态信息。
- 新闻管理功能：新闻数据是网站的主要数据，要求后台数据库存放各条新闻的相关



信息，形成新闻的动态管理。

- 栏目管理功能：将新闻分成若干类别，形成网站不同的栏目。要求后台存放栏目的相关信息。
- 专题管理功能：可能会就某个专题进行新闻讨论，需要按专题对新闻进行组织，形成不同专题的新闻条目。要求后台存放专题的相关信息。

（2）标识实体



知识储备

实体（Entity）：是指客观存在而又可以相互区别的事物，该事物具有可区分于他物的特征或属性，并与其他实体有一定的联系。实体可以是具体的事物，例如一名注册用户，一条新闻等。

实体集：性质相同的同类实体的集合称为实体集。例如所有的注册用户，所有的新闻等。

属性：实体的特征称为实体的属性。每个实体都可以有很多特性，每一个特性称为一个属性。每个属性有一个值域，其类型可以是整数型、实数型、日期型、文本型等。例如注册用户有用户编号、用户姓名、出生日期、性别等属性，新闻有新闻标题、新闻内容、新闻作者、新闻来源、新闻发表时间等属性。

实体标识符：能唯一标识实体的属性或属性集称为实体标识符。有时也称为关键码、键或主键。例如注册用户的用户编号可作为注册用户实体的标识符。

关系（Relationship）：两个或多个实体之间的联系。例如某个用户与某条新闻之间的管理关系。

关系集：相同类型的关系的集合。例如所有用户与所有新闻之间的管理关系。

“新闻类别网站”的实体有以下几个。

- 用户实体：在“新闻类别网站”中具有新闻编辑功能的用户。
- 新闻文章实体：“新闻类别网站”中的新闻文章。
- 日志实体：记录“新闻类别网站”被用户操作的各项活动。
- 栏目实体：“新闻类别网站”的网站栏目。
- 专题实体：“新闻类别网站”的新闻分类专题。

（3）标识每个实体的属性

实体的属性即实体需要存储的详细信息。

用户实体的属性有：用户名、用户密码、用户类型、用户注册时间、最后登录时间、最后登录 IP、用户发表的新闻数、用户的登录次数。

新闻实体的属性有：新闻文章标题、文章导读信息、文章内容、文章关键字、文章作者、文章来源、文章添加时间、文章相关的图片、文章所属栏目、文章所属专题、操作用户名、文章的点击次数。

日志实体的属性有：操作用户名、日志类型、日志添加时间、操作用户的 IP 地址、操作结果。

栏目实体的属性有：栏目名称、栏目的英文名称、栏目简介、栏目 URL、栏目添加时间、

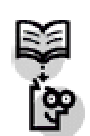


栏目排序情况、栏目的父栏目、栏目是否隐藏。

专题实体的属性有：专题名称、专题简介、专题添加时间。

(4) 标识实体之间的关系

现实中的这些实体，在逻辑上具有一定的联系。例如用户和新闻两个实体，用户实体对新闻实体做管理，用户实体可以添加新闻、编辑新闻、删除新闻。需要对前面所列的实体进行分析，寻找各实体之间的联系，确定各实体之间的逻辑关系，以便于在数据库中实现这些联系，确保数据库中反映的数据信息和现实中的实体间的关系是相互一致的。



知识储备

实体间的关系有一对一关系、一对多关系、多对多关系。

- 一对一关系：如果实体集 E1 中每个实体最多和实体集 E2 中的一个实体有关系，反之亦然，那么实体集 E1 和 E2 的关系称为“一对一关系”，记为“1∶1”。
- 一对多关系：如果实体集 E1 中每个实体可以与实体集 E2 中任意一个(零个或多个)实体间有关系，而 E2 中每个实体最多和 E1 中一个实体有关系，那么称 E1 对 E2 的关系是“一对多关系”，记为“1∶n”。
- 多对多关系：如果实体集 E1 中每个实体可以与实体集 E2 中任意一个(零个或多个)实体有关系，反之亦然，那么称 E1 和 E2 的关系是“多对多关系”，记为“m∶n”。

“新闻类别网站”的实体间关系描述如下：

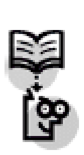
① 用户实体管理新闻实体，一个用户可以管理多条新闻，一条新闻只能被一个用户管理。这二者之间的关系可以描述为一对多的关系。用户表称为主表，新闻表称为从表。

② 系统日志记录用户操作系统的情况，用户操作网站一次，将被记录一次，日志可以记录所有用户的操作情况。日志表称为主表，用户表称为从表。

③ 系统栏目的每个栏目可以包括多条新闻，而每条新闻只能属于一个栏目。栏目表称为主表，新闻表称为从表。

④ 系统专题的每个专题可以包括多条新闻，而每条新闻只能属于一个专题。专题表称为主表，新闻表称为从表。

(5) 绘制实体—关系(E-R，Entity-Relationship)图



知识储备

E-R 图中的符号有以下几种。

- 矩形框：表示实体集。
- 菱形框：表示实体间的关系集。
- 椭圆形框：表示实体或关系的属性。
- 连线：连接属性和实体集，或者连接实体集和关系集，实体集与实体集间的映射基数标注在连线的附近。

根据上面的 E-R 图符号定义，可以绘制出“新闻类别网站”的 E-R 图，如图 3-1 所示。

(6) 将 E-R 图转化为表

① 将各实体转化为对应的表，每个实体对应一张表；将各属性转化为各表对应的列，每





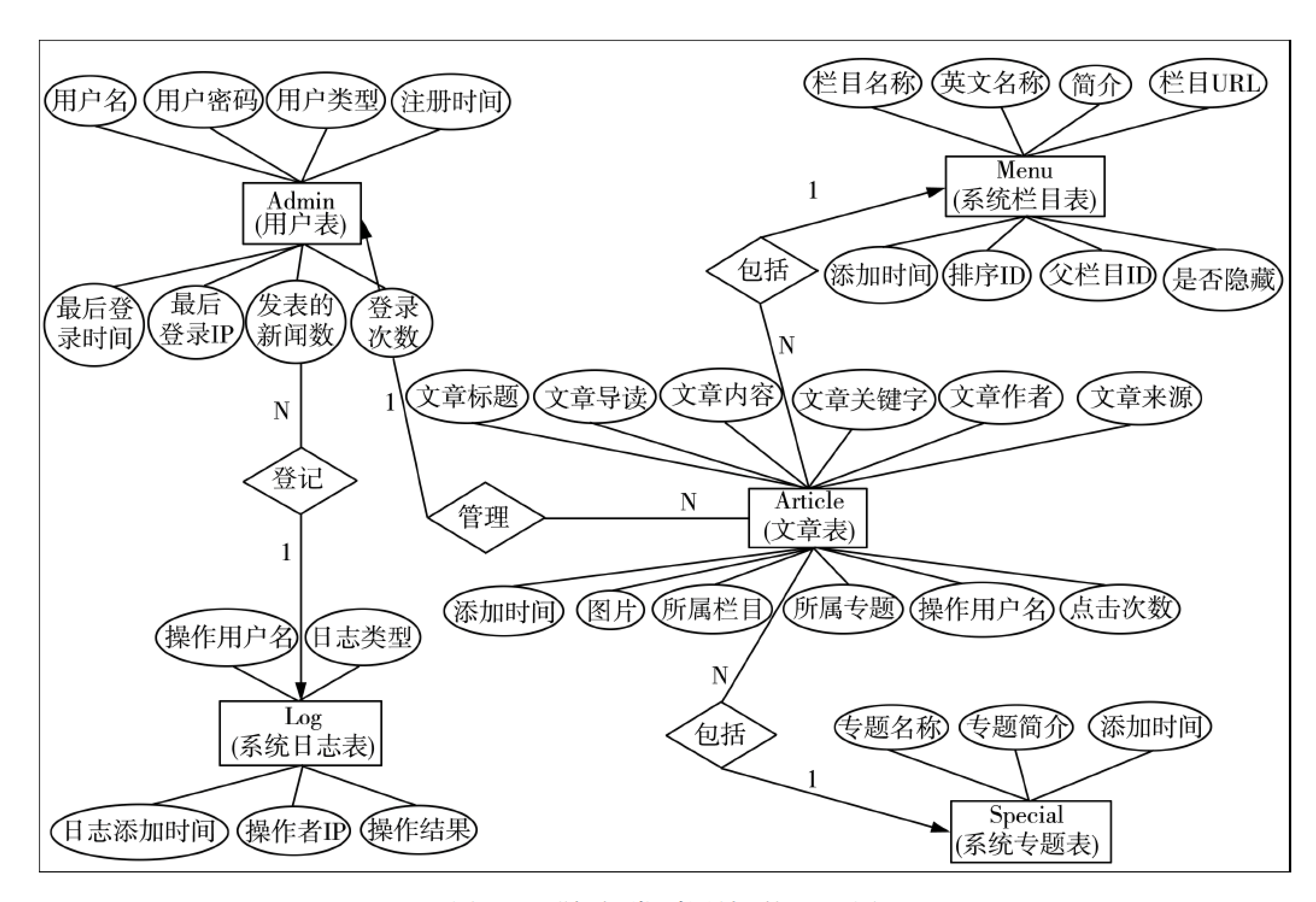


图 3-1　“新闻类别网站”的 E-R 图

个属性对应一个列。

② 标识各表的主键列，若该表没有主键列，添加 ID(编号)列，ID 列没有实际的含义，只是用来唯一标识该表的记录，即作为该表的主键列。

③ 将实体间的映射关系转化为表间的主键、外键关系。

转化后的结果如表 3-1～表 3-5 所示。

表 3-1　Admin（用户表）

字　段　名	数 据 类 型	说　　明
[ID]	[int]	用户编号，自动产生
[AdminName]	[varchar(50)]	用户名，主键
[AdminPwd]	[varchar(50)]	用户密码，不能为空
[AdminType]	[smallint]	用户类型，不能为空
[AddTime]	[smalldatetime]	注册时间
[LastLoginTime]	[smalldatetime]	最后登录时间
[LastLoginIP]	[varchar(50)]	最后登录 IP
[NewsNum]	[int]	发表的新闻数
[LoginNum]	[int]	登录次数



表 3-2　Article（文章表）

字　段　名	数 据 类 型	说　　明
[ID]	[int]	文章编号，主键，自动产生
[ArticleTitle]	[varchar(255)]	文章标题，不能为空
[ArticleInfo]	[varchar(255)]	文章导读
[ArticleContent]	[ntext]	文章内容
[ArticleKey]	[varchar(255)]	文章关键字
[ArticleAuthor]	[varchar(50)]	文章作者
[ArticleFrom]	[varchar(255)]	文章来源
[ArticleTime]	[smalldatetime]	文章添加时间
[ArticlePicture]	[varchar(255)]	文章中的图片存储路径
[MenuID]	[varchar(50)]	文章所属栏目，外键
[IsHead]	[int]	是否置顶
[IsImg]	[int]	是否图片新闻
[IsHot]	[int]	是否热点新闻
[SpecialID]	[varchar (50)]	文章所属专题，外键
[AdminName]	[varchar(50)]	文章操作用户，外键
[Hit]	[int]	点击次数

表 3-3　Log（系统日志表）

字　段　名	数 据 类 型	说　　明
[LogID]	int	日志 ID，主键，自动产生
[AdminName]	[varchar(50)]	操作用户名，外键
[LogType]	[varchar(100)]	日志类型
[LogTime]	[datetime]	日志添加时间
[IP]	[varchar(50)]	操作者 IP
[Result]	[varchar(3)]	操作结果

表 3-4　Menu（系统栏目表）

字　段　名	数 据 类 型	说　　明
[MenuID]	[varchar(50)]	栏目 ID，主键
[MenuName]	[varchar(50)]	栏目名称，不能为空
[MenuEName]	[varchar(50)]	栏目的英文名称
[MenuMe]	[varchar(200)]	栏目简介

续表

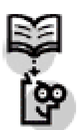
字 段 名	数 据 类 型	说 明
[MenuURL]	[varchar(200)]	栏目 URL
[AddTime]	[datetime]	栏目添加时间
[OrderID]	[int]	栏目排序 ID
[ParentID]	[int]	父栏目 ID
[IsHide]	[int]	是否隐藏

表 3-5 Special（系统专题表）

字 段 名	数 据 类 型	说 明
[SpecialID]	[varchar(50)]	专题 ID，主键
[SpecialName]	[varchar(50)]	专题名称
[SpecialInfo]	[varchar(100)]	专题简介
[SpecialTime]	[datetime]	专题添加时间

（7）数据规范化

接下来是对上面形成的表进行规范化，数据规范化应该按照规范化标准来进行。



知识储备

数据规范化是要消除表中的数据冗余和操作异常。数据规范化的标准按照第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、第四范式(4NF)、第五范式(5NF)来进行，对于绝大多数的实际应用来说，3NF 已能够消除大多数普通情况下的数据冗余和操作异常。

使数据达到第三范式，可以按照下面的方法来做：

① 删除表中重复的列以达到第一范式。

② 删除表中不完全依赖于主键的数据以获得第二范式。

③ 删除不属于该表的数据（即完全依赖于其他列或外键的数据项），使数据符合第三范式。

④ 检查表的每一行是否表示有意义的信息。

⑤ 允许表中出现一定数量的冗余数据，以减少操作数据时的跨表连接，从而提高系统性能。这种增加数据冗余以提高数据库性能的过程称为非规范化。

用 3NF 的要求检查“新闻类别网站”的表结构，可以发现这些表的定义已满足 3NF。

（8）将 E-R 图转换成数据库的逻辑模型（如图 3-2 所示）

（9）备注

为了方便网站的编程与后期维护管理，在上述的业务数据表之外，另增加两个数据表，一个是系统配置表，存放系统的一些配置信息，如表 3-6 所示；另一个是超链接表，存放与本网站有链接的其他网站的信息，如表 3-7 所示。



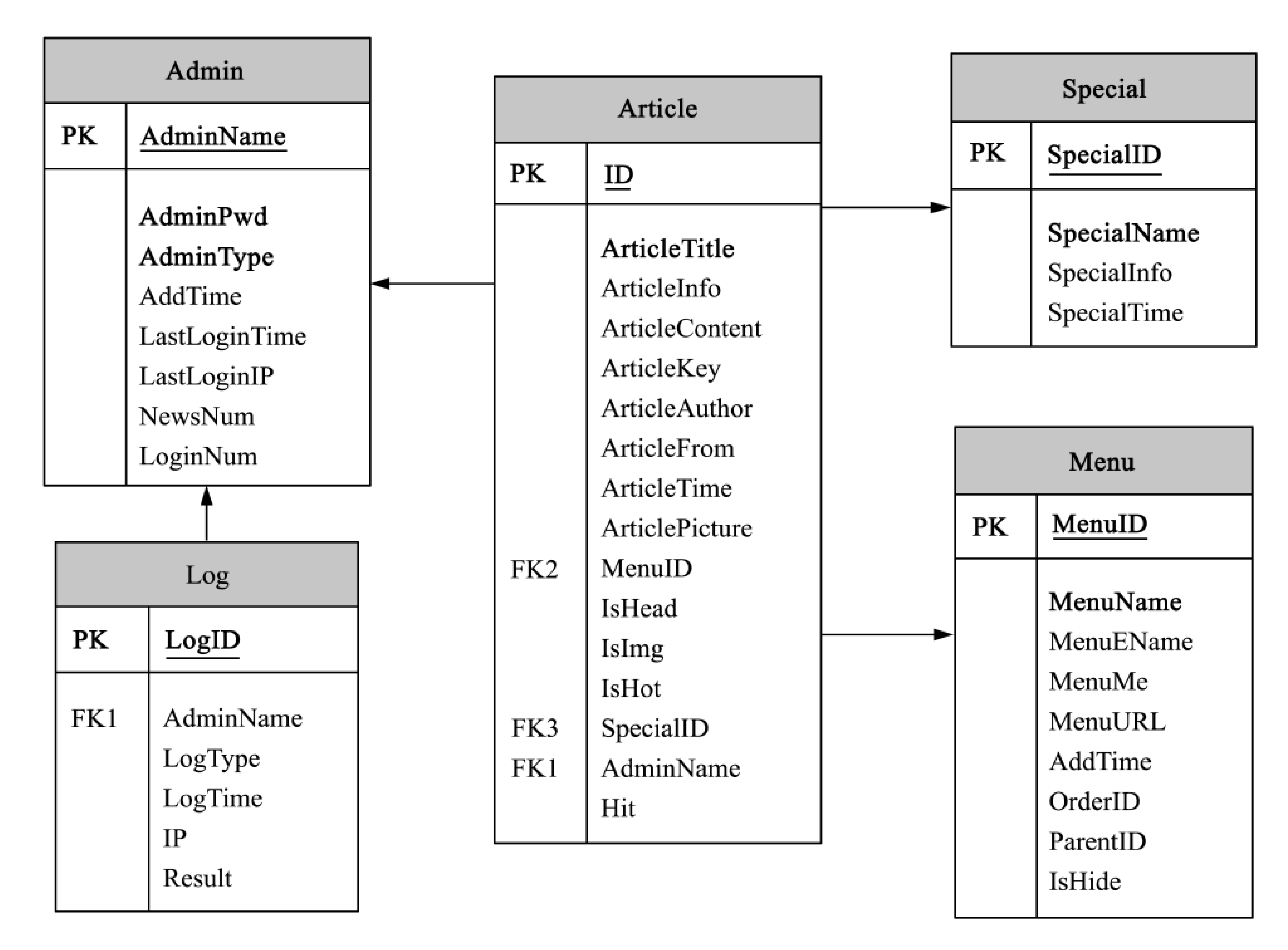


图 3-2 “新闻类别网站”的数据库逻辑模型图

表 3-6 Config（系统配置表）

字 段 名	数 据 类 型	说 明
[configID]	[int]	配置 ID，主键
[Title]	[varchar(50)]	系统名称
[Author]	[varchar(50)]	系统作者
[Description]	[varchar(50)]	系统描述
[Keyword]	[varchar(50)]	系统关键字
[Copyright]	[varchar(50)]	版权信息
[Email]	[varchar(50)]	E-mail 地址
[AdminUserListNum]	[int]	后台用户列表数
[AdminLogListNum]	[int]	后台日志列表数
[AdminNewsListNum]	[int]	后台文章列表数
[HotNewsNum]	[int]	热点新闻列表数
[HeadNewsNum]	[int]	头条新闻列表数
[TopNewsNum]	[int]	最新新闻列表数
[TopImgNum]	[int]	最新图片列表数
[ClassNewsNum]	[int]	栏目新闻列表数

表 3-7　Link（超链接表）

字　段　名	数 据 类 型	说　　明
[LinkID]	[numeric]	超链接 ID,主键
[SiteName]	[varchar(50)]	站点名称
[Readme]	[varchar(50)]	站点简介
[URL]	[varchar(100)]	URL
[Orders]	[int]	排序 ID

3.1.3　相关知识拓展

本节的 E-R 图和数据库模型图均是使用 Microsoft Office Visio Professional 2003 绘制的。请先在计算机上安装该软件,然后参照下面的方法绘制上述的图形。

1. E-R 图的绘制

绘制 E-R 图的具体操作步骤如下:

(1) 打开 Microsoft Office Visio 2003,在【选择绘图类型】中选择【框图】|【基本框图】模板,如图 3-3 所示。

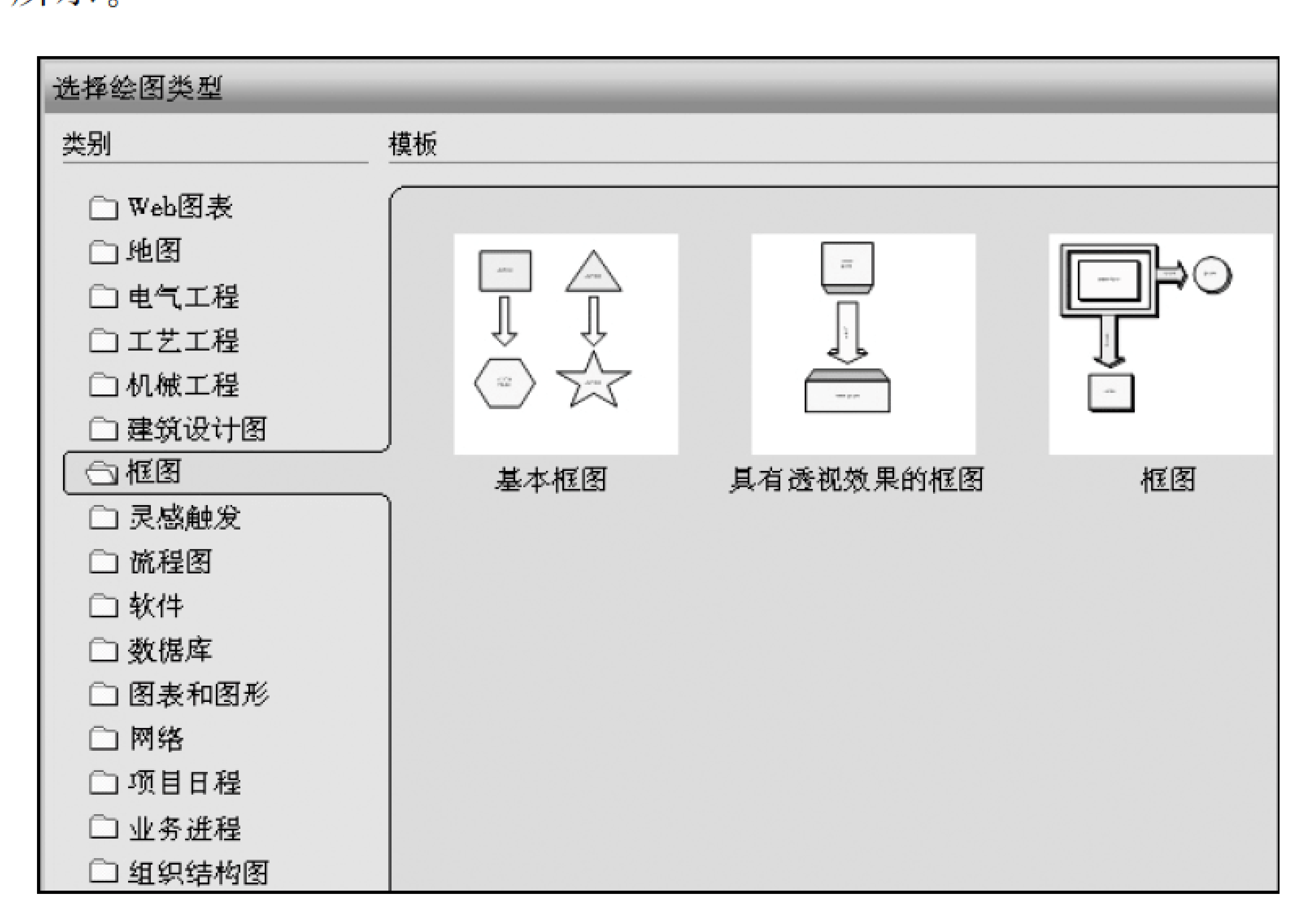


图 3-3　选择绘图类型

(2) 在打开的工作界面中,从【形状】面板的【基本形状】中选择"矩形"、"椭圆"。若看不到"菱形",可利用【形状】面板的【搜索形状】下拉列表进行搜索,如图 3-4 所示。



图 3-4　搜索形状

绘制线条时,可以选择【视图】菜单下的【工具栏】|【绘图】命令,再使用【绘图】工具栏中的直线进行绘制。

将所需要的形状拖动至绘图页,双击该对象可进行文字设



置。更多的其他操作请读者自行学习。

2. 数据库模型图的绘制

绘制数据库模型图的具体操作步骤如下：

（1）单击【文件】|【新建】|【数据库】|【数据库模型图】命令，得到的效果如图 3-5 所示。

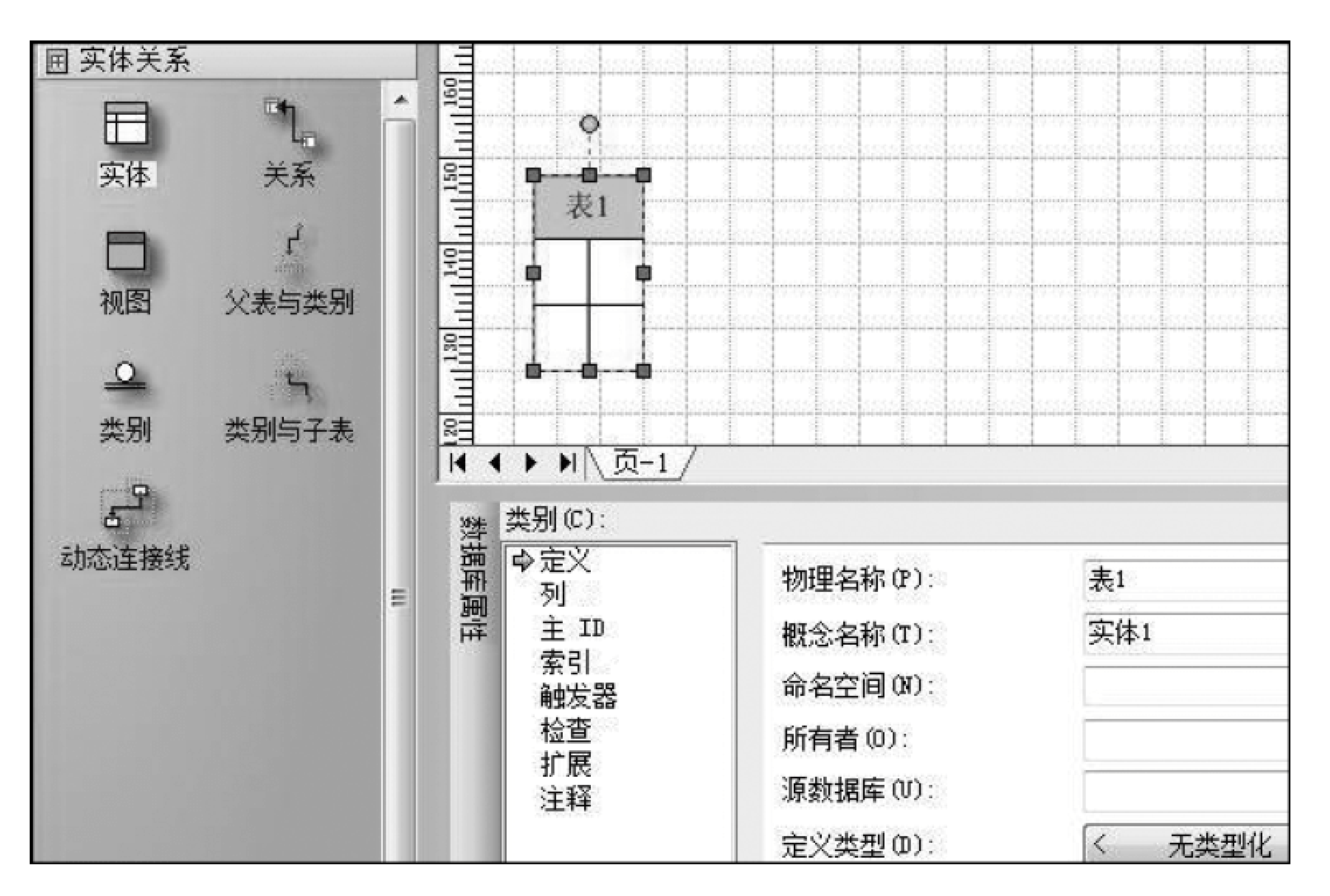


图 3-5　新建数据库模型图

（2）拖动【实体】图标至右侧的绘图页，松开鼠标出现一个新的实体，在绘图页下面的【数据库属性】面板中对该实体进行各种设置，如图 3-6 所示。



图 3-6　实体的设置

（3）以用户实体的设置为例，图 3-7 是设置好的用户实体。

下面进行用户实体的具体设置。

①【定义】页面的详细设置如图 3-8 所示。

②【列】页面的详细设置如图 3-9 所示。



Admin	
PK	AdminName
	AdminPwd
	AdminType
	AddTime
	LastLoginTime
	LastLoginIP
	NewsNum
	LoginNum

图 3-7　用户实体模型图

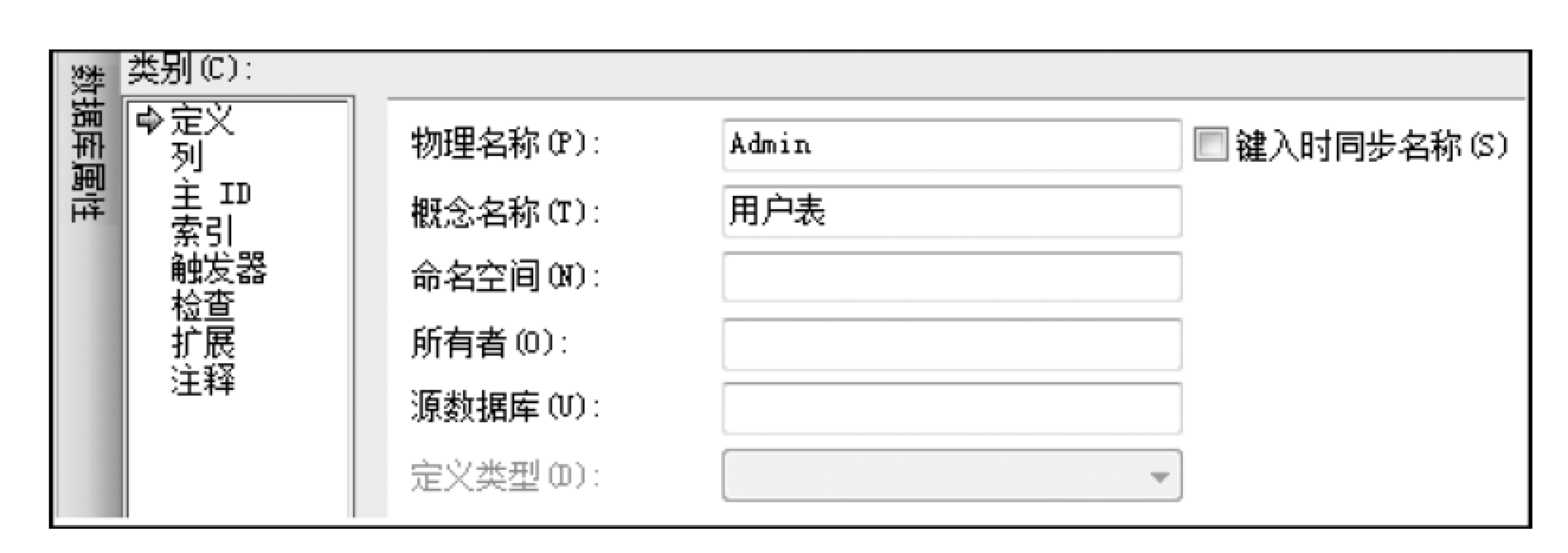
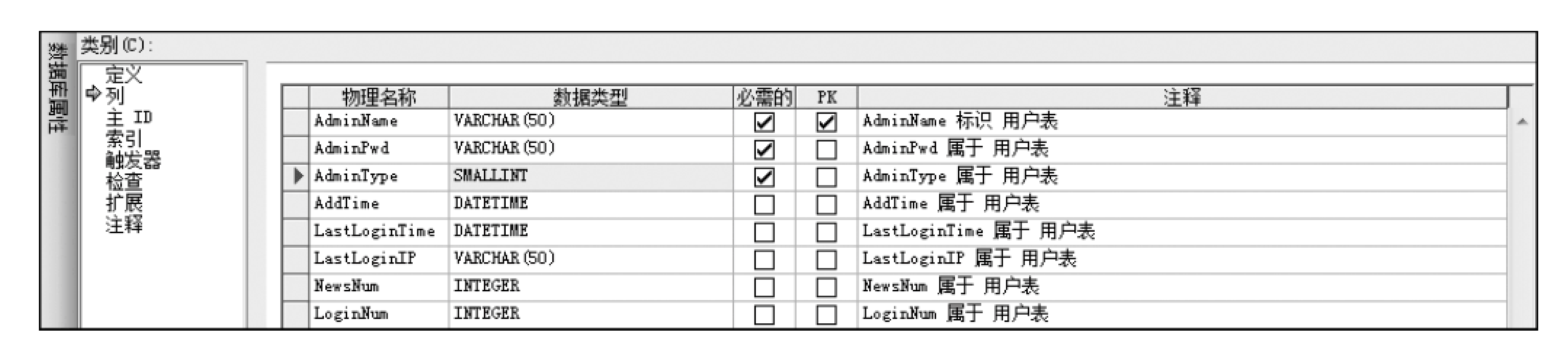


图 3-8　【定义】页面的设置



物理名称	数据类型	必需的	PK	注释
AdminName	VARCHAR(50)	☑	☑	AdminName 标识 用户表
AdminPwd	VARCHAR(50)	☑	☐	AdminPwd 属于 用户表
AdminType	SMALLINT	☑	☐	AdminType 属于 用户表
AddTime	DATETIME	☐	☐	AddTime 属于 用户表
LastLoginTime	DATETIME	☐	☐	LastLoginTime 属于 用户表
LastLoginIP	VARCHAR(50)	☐	☐	LastLoginIP 属于 用户表
NewsNum	INTEGER	☐	☐	NewsNum 属于 用户表
LoginNum	INTEGER	☐	☐	LoginNum 属于 用户表

图 3-9　【列】页面的设置

其他的页面不必设置。

按照上面的方法创建其他的实体。然后拖动图 3-5 左侧所示的【关系】图标至主表，当主表周围出现红色的边框时放开鼠标左键；再拖动这条“关系”线的尾部至从表，当从表周围出现红色的边框时放开鼠标左键，此时两个表的关系即可成功建立。这条“关系”线的箭头方向指向主表。

3.1.4　经验总结

对于 E-R 图的学习，初学者一般不甚理解，但这个方法确实是确定未来系统的数据表及其结构的好方法，读者可以多读相关的数据库设计方面的书籍，以利于更好地使用该方法。

3.2　创建数据库

3.2.1　任务描述

在 SQL Server 2005 中创建“新闻类别网站”的数据库。

3.2.2　任务实现

关于 Microsoft SQL Server 2005 软件的安装请参见 5.3 节。



创建“新闻类别网站”数据库的具体操作步骤如下：

(1) 打开 SQL Server 2005|SQL Server Management Studio，链接至数据库服务器，右击【对象资源管理器】面板上的【数据库】对象，弹出如图 3-10 所示的快捷菜单，选择【新建数据库】命令，打开如图 3-11 所示的【新建数据库】对话框。

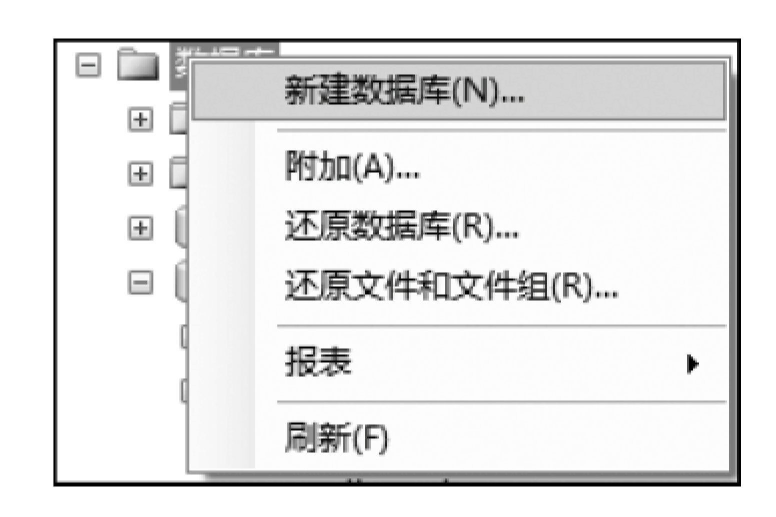


图 3-10　数据库的快捷菜单

(2) 在【数据库名称】文本框中输入“新闻类别网站”的数据库名 News，修改【数据库文件】列表，将列表中第一行的【逻辑名称】文本框值改为 News_Data，第二行不用修改(实际情况下可根据需要进行修改)，单击【确定】按钮。在【对象资源管理器】面板上的【数据库】对象中即可看到新建的 News 数据库。

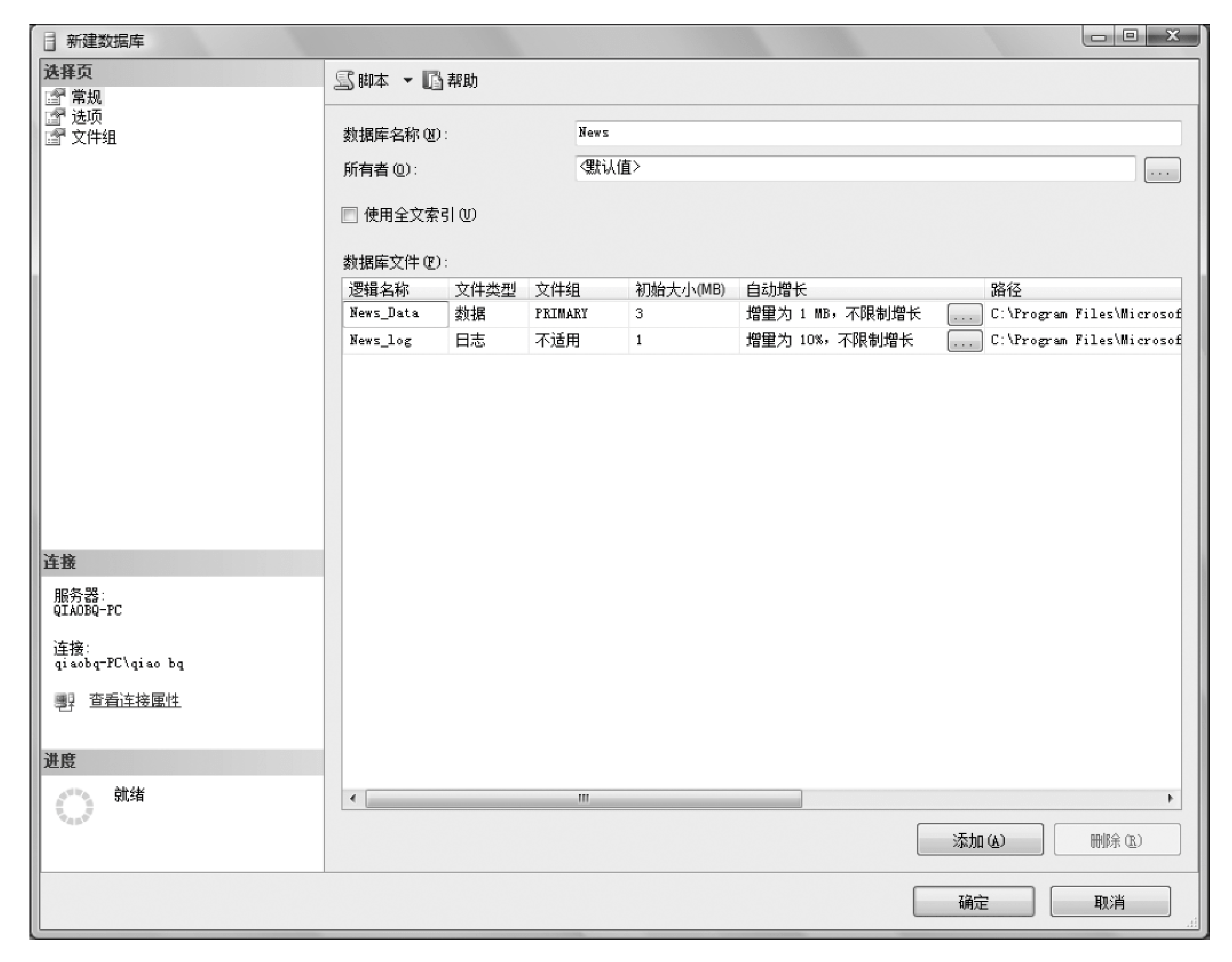


图 3-11　【新建数据库】对话框

3.2.3　相关知识拓展

在 SQL Server 2005 中，数据库是所有对象的集合体，它包括的对象有表、视图、存储过程、触发器等。网站的后台数据需要存在数据库中，而且是存储在数据库的表对象中。

前面创建的数据库 News 是一种逻辑上的数据库，在 SQL Server 中，数据库用户是在





逻辑层面上使用数据，这使他们不需要过多了解复杂知识就可以使用数据库。

不论什么形式的数据，在操作系统的管理方式下，均是以文件的形式进行组织。数据库 News 在物理磁盘上也是以文件的形式进行组织的。

逻辑数据库 News 对应两个磁盘文件，一个文件名是“News_Data.mdf”，另一个是“News_Log.ldf”。这说明在使用逻辑数据库 News 时，其实 SQL Server 2005 真正在操作的是这两个物理文件，或者说在操作存储在物理磁盘上的这两个文件。

下面进一步说明数据库的物理文件，理解起来稍微有一些难度。

实际上，每个 SQL Server 2005 的逻辑数据库最少对应两个物理磁盘文件，创建数据库时，默认方式下系统将创建这两个物理磁盘文件。这一点从 News 数据库可以看出（News 数据库是按照默认方式创建的数据库）。这两个文件，一个称为主要数据文件，扩展名是 .mdf；另一个称为日志文件，扩展名是 .ldf。比如 News 数据库的 News_Data.mdf 和 News_Log.ldf，这两个文件是必不可少的。

但是，SQL Server 2005 中一个逻辑数据库可以对应多个物理磁盘文件。不论有多少物理磁盘文件，都可以把这些文件分为两大类，一类是数据文件；一类是日志文件。对于数据文件，又可以分为主要数据文件（扩展名为 .mdf）、次要数据文件（扩展名为 .ndf）。

细心的读者们，能不能看出 News 数据库的 News_Data.mdf 文件就是主要数据文件呢？一个数据库有且仅能有一个主要数据文件，但可以根据需要有多个次要数据文件。当以默认方式创建数据库时，主要数据文件就是系统自动创建的两个物理磁盘文件之一，但是系统没有创建次要数据文件。所以次要数据文件不是必须的。

数据库可以没有次要数据文件，也可以有多个次要数据文件。可以把这些数据文件存放在不同的磁盘上，这就需要在创建数据库时输入各个数据文件的路径。

日志文件是 SQL Server 2005 用来自动记录数据库的操作日志的。在数据库中，日志文件包含恢复数据库所需的所有日志信息。每个数据库必须至少有一个日志文件，当以默认方式创建数据库时，日志文件也是系统自动创建的两个物理磁盘文件之一。但实际上日志文件可以不止一个。

为了更好地实现数据库文件的组织，SQL Server 引入了文件组（Filegroup）的概念。

每个数据库有一个主要文件组（Primary），此文件组包含主要数据文件和未放入其他文件组的所有次要数据文件。可以创建用户定义的文件组，用于将数据文件集合起来，以便于管理、分配数据。通过设置文件组，可以有效地提高数据库的读写性能。本项目中未设置新的文件组。更多的资料请查看相关书籍或者 SQL Server 2005 的帮助文档。

当需要删除数据库时，请在【对象资源管理器】面板中选中需要删除的数据库，右击鼠标，在弹出的快捷菜单中选择【删除】命令。

3.2.4　经验总结

创建数据库的操作界面有许多选项，可以根据需要来进行设置。在本项目的操作中，有一些注意事项，说明如下。

在设置数据文件或者日志文件的存储路径时，该路径必须是已存在的路径，若输入不存在的路径，则创建会出错。例如图 3-12 中所示，创建 News 数据库时将 News_Data.mdf 主



要数据文件存储在 data 文件夹下，但创建(CREATE DATABASE)却失败。

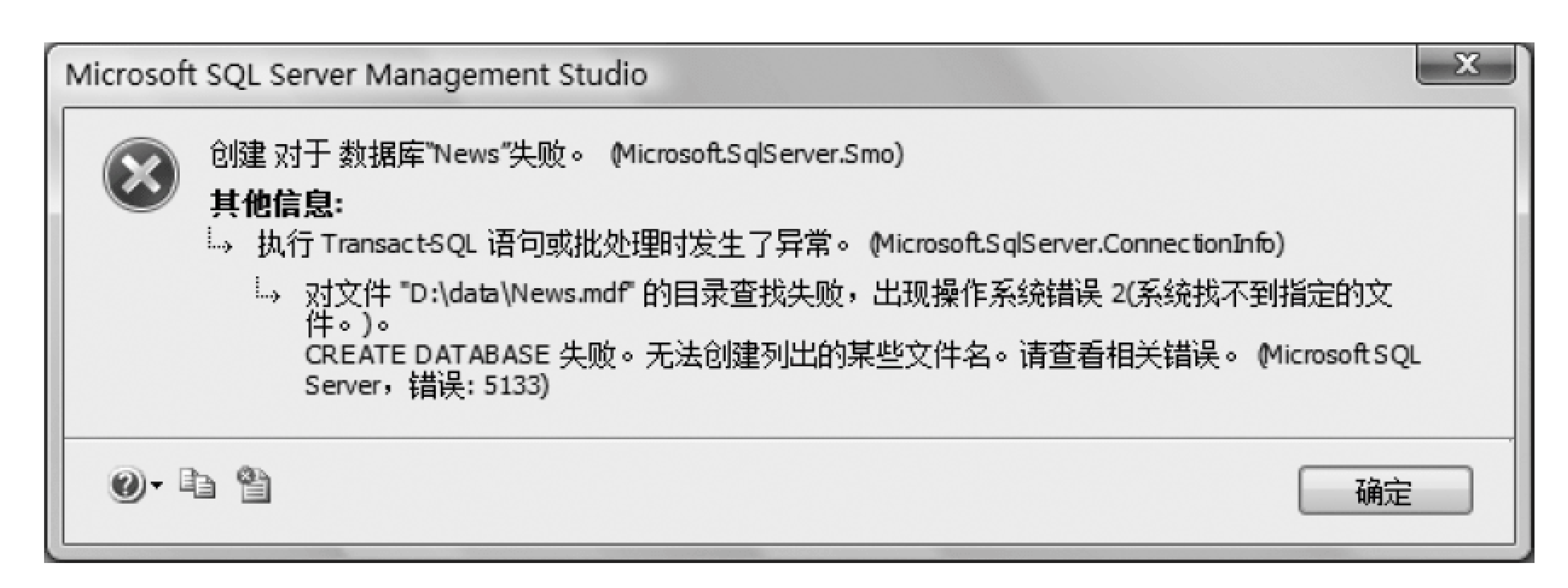


图 3-12　创建出错时的对话框

另外，创建数据库时，默认情况下，系统会将数据文件和日志文件放在同一个驱动器上的同一个文件夹下。这是为处理单磁盘系统而采用的方法。在生产环境中，这可能不是最佳的方法。建议将数据文件和日志文件放在不同的磁盘上。

3.3　创　建　表

知识储备

表是数据库中最重要的对象，用来存储数据。数据在表中的组织方式与在电子表格中相似，都是按行和列的格式组织的。每一行代表表中一条唯一的记录，每一列代表记录中的一个字段。例如，如图 3-13 所示的 Admin 表包含 5 个列，即 5 个字段；有 7 行数据，即 7 条记录。每条记录代表一位管理员的相关信息，这些信息由 5 个列进行说明，并分别存储在这 5 个列中。

ID	AdminName	AdminPwd	AdminType	AddTime
36	admin	F59BD65F7EDAFB087A81D4DCA06C4910	2	2008/8/15 15:02:00
16	afffmn1	CB939D9EA1198112BF8EDCD343ED2A1D	3	2008/8/5 0:00:00
31	afffmn2	CB939D9EA1198112BF8EDCD343ED2A1D	3	2008/8/5 0:00:00
5	ain1	CB939D9EA1198112BF8EDCD343ED2A1D	2	2008/8/5 0:00:00
20	ain2	CB939D9EA1198112BF8EDCD343ED2A1D	3	2008/8/5 0:00:00
6	amn1	CB939D9EA1198112BF8EDCD343ED2A1D	2	2008/8/5 0:00:00
21	amn2	CB939D9EA1198112BF8EDCD343ED2A1D	3	2008/8/5 0:00:00

图 3-13　Admin 表

在 SQL Server 2005 中，每个表包括两个部分，即表的列定义和表的记录。对于每个表，先定义其包含的列，再往表中输入记录。表的列定义也称为表结构定义，相当于在一张空白的纸上先画出表的框架来，这个框架包括表分为几个列、每个列占多宽(列的宽度)、列名是什么(列名)、计划在每列中填入什么样的数据(即填入的数据具有什么样的取值特征，是一个日期还是一个有数学意义的数值还是一串字符等，这称为列的数据类型)，这个框架就称为表的结构。表中的记录通常由软件使用者利用软件提供的输入界面进行输入，较少情况由网站开发人员进行预先输入。





3.3.1　任务描述

在 SQL Server 2005 中创建“新闻类别网站”的各个表。

3.3.2　任务实现

“新闻类别网站”共有 7 个表，下面以 Article 表的建立为例，说明本阶段任务实现的步骤，其他的表由读者按此步骤自己练习进行创建。

(1) 创建表

建立 Article 表的具体操作步骤如下：

① 单击 News 数据库前面的加号，使其变成减号，展开该数据库。可以看到其中含有【表】这个对象，在【表】上右击，弹出如图 3-14 所示的快捷菜单，选择该菜单上【新建表】命令，在打开的界面中进行表的结构定义（新建表就是让网站开发人员进行表结构的定义，不是用来输入记录数据的）。



图 3-14　新建表

② 如图 3-15 所示，在【列名】下的文本框中输入 Article 表的第一个列名 ID，在【数据类型】中选择 ID 列的数据类型（关于数据类型的内容参见 3.3.3 小节的相关知识拓展），在【允许空】中设置 ID 列的取值是否可以为空，打对号表示允许空值，不打对号表示不允许空值。类似地，再依次输入 Article 表的其他列的定义。

③ Article 表中的 ID 列是主键，其取值为自动增长。要将某个列设为主键，可在该列上右击，选择快捷菜单上的【设置主键】命令，如图 3-16 所示，可以看到该列被标上钥匙图案，即为该表的主键。一个表只能有一个主键。

QIAOBQ-PC.News - dbo.Article*

列名	数据类型	允许空
ID	int	□
ArticleTitle	varchar(255)	□
ArticlesInfo	varchar(255)	☑
ArticleContent	text	☑
ArticleKey	varchar(255)	☑
ArticleAuthor	varchar(50)	☑
ArticleFrom	varchar(255)	☑
ArticleTime	smalldatetime	☑
ArticlePicture	varchar(255)	☑
MenuID	varchar(50)	☑
IsHead	int	☑
IsImg	int	☑
IsHot	int	☑
SpecialID	varchar(50)	☑
AdminName	varchar(50)	☑
Hit	int	☑

图 3-15　Article 表



图 3-16　设置主键





④ 要将某个列的取值设为自动增长，请选择该列，在其对应的【列属性】面板的【标识规范】中进行设置，如图 3-17 所示。其中【标识增量】表示下条记录比上条记录自动增长的幅度，这里输入值“1”；【标识种子】表示第一条记录的自动增长列的初始值，这里输入值也是“1”。

⑤ 录入完成后单击 SQL Server 2005 工具栏上的存盘标志，出现如图 3-18 所示的对话框，输入表的名称 Article，单击【确定】按钮，在数据库 News 中便出现了该 Article 表对象。

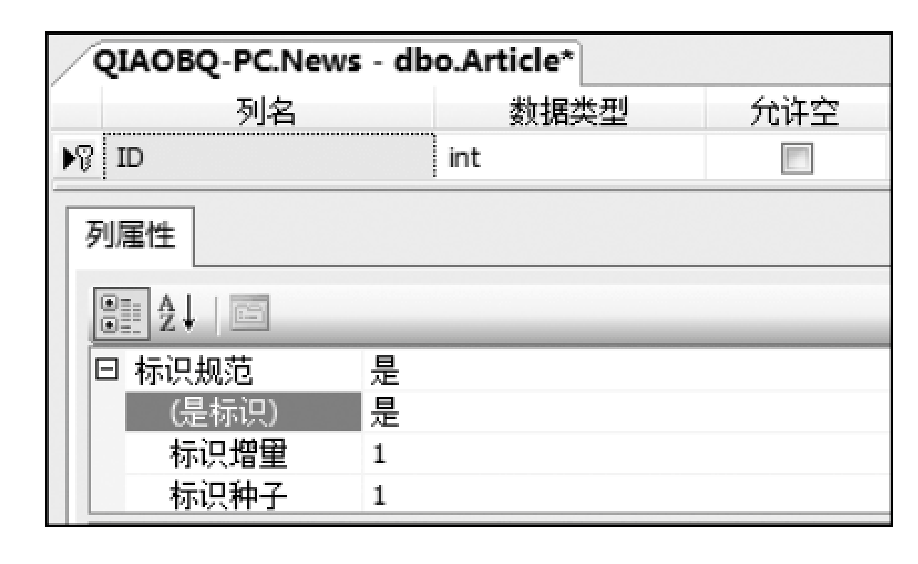



图 3-17　设置【列属性】



图 3-18　保存表

News 数据库中其他表由读者按照上面的步骤自行创建。

其他表结构分别如图 3-19～图 3-24 所示。

QIAOBQ-PC.News - dbo.Admin*

列名	数据类型	允许空
ID	int	☐
AdminName	varchar(50)	☐
AdminPwd	varchar(50)	☐
AdminType	smallint	☐
AddTime	datetime	☑
LastLoginTime	datetime	☑
LastLoginIP	varchar(50)	☑
NewsNum	int	☑
LoginNum	int	☑

图 3-19　Admin 表

QIAOBQ-PC.News - dbo.Log*

列名	数据类型	允许空
LogID	int	☐
AdminName	varchar(50)	☑
LogType	varchar(100)	☑
LogTime	datetime	☑
IP	varchar(50)	☑
Result	varchar(3)	☑

图 3-20　Log 表

QIAOBQ-PC.News - dbo.Menu*

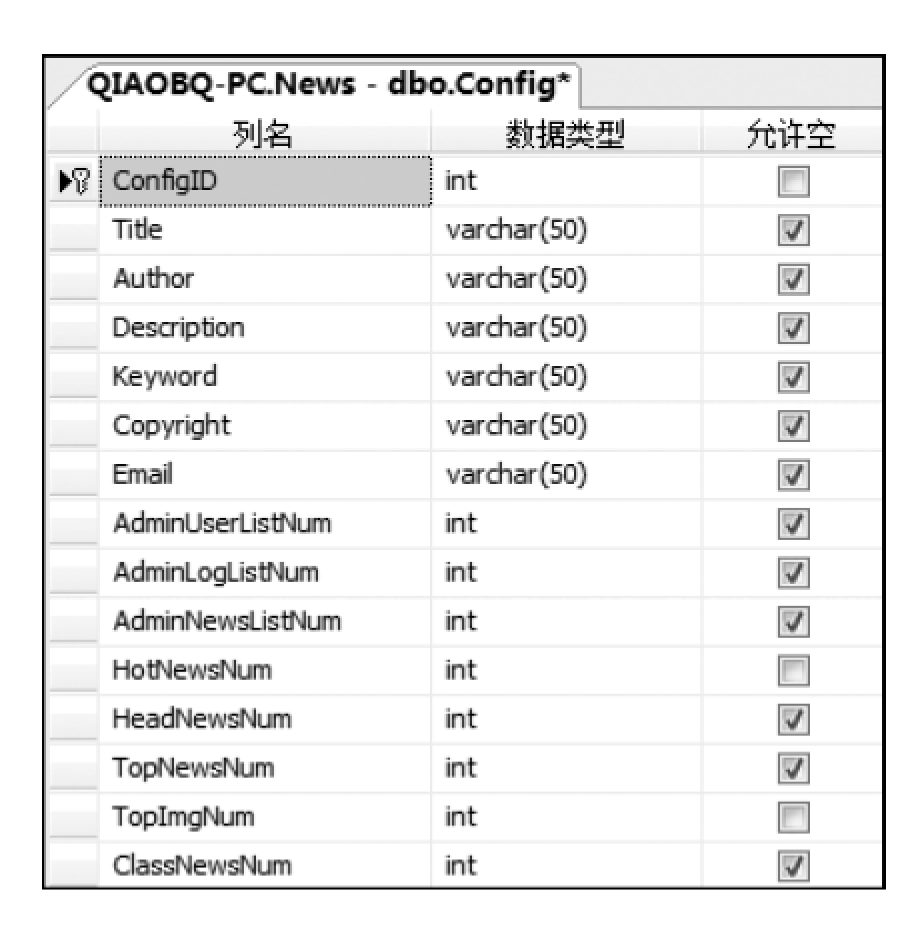
列名	数据类型	允许空
MenuID	varchar(50)	☐
MenuName	varchar(50)	☐
MenuEName	varchar(50)	☑
MenuMe	varchar(200)	☑
MenuURL	varchar(200)	☑
AddTime	datetime	☑
OrderID	int	☑
ParentID	int	☑
IsHide	int	☑

图 3-21　Menu 表

QIAOBQ-PC.News - dbo.Special

列名	数据类型	允许空
SpecialID	varchar(50)	☐
SpecialName	varchar(50)	☑
SpecialINfo	varchar(100)	☑
SpecialTime	datetime	☑

图 3-22　Special 表


QIAOBQ-PC.News - dbo.Config*

列名	数据类型	允许空
ConfigID	int	☐
Title	varchar(50)	☑
Author	varchar(50)	☑
Description	varchar(50)	☑
Keyword	varchar(50)	☑
Copyright	varchar(50)	☑
Email	varchar(50)	☑
AdminUserListNum	int	☑
AdminLogListNum	int	☑
AdminNewsListNum	int	☑
HotNewsNum	int	☐
HeadNewsNum	int	☑
TopNewsNum	int	☑
TopImgNum	int	☐
ClassNewsNum	int	☑

图 3-23　Config 表





（2）创建表间关系

各表创建完成后，要设置表间关系，将主表和从表间的关系反映在数据库中，使各表间的数据能够反映实际情况下的逻辑依赖关系。在数据库中，主从关系由主表的主键和从表的外键反映，设置方法有多种，下面介绍其中的一种。

① 在【对象资源管理器】面板中展开 News 数据库，右击【数据库关系图】对象，选择快捷菜单中的【新建数据库关系图】命令，如图 3-25 所示。

QIAOBQ-PC.News - dbo.Link

列名	数据类型	允许空
LinkID	int	☐
SiteName	varchar(50)	☑
ReadMe	varchar(50)	☑
URL	varchar(100)	☑
Orders	int	☑

图 3-24　Link 表

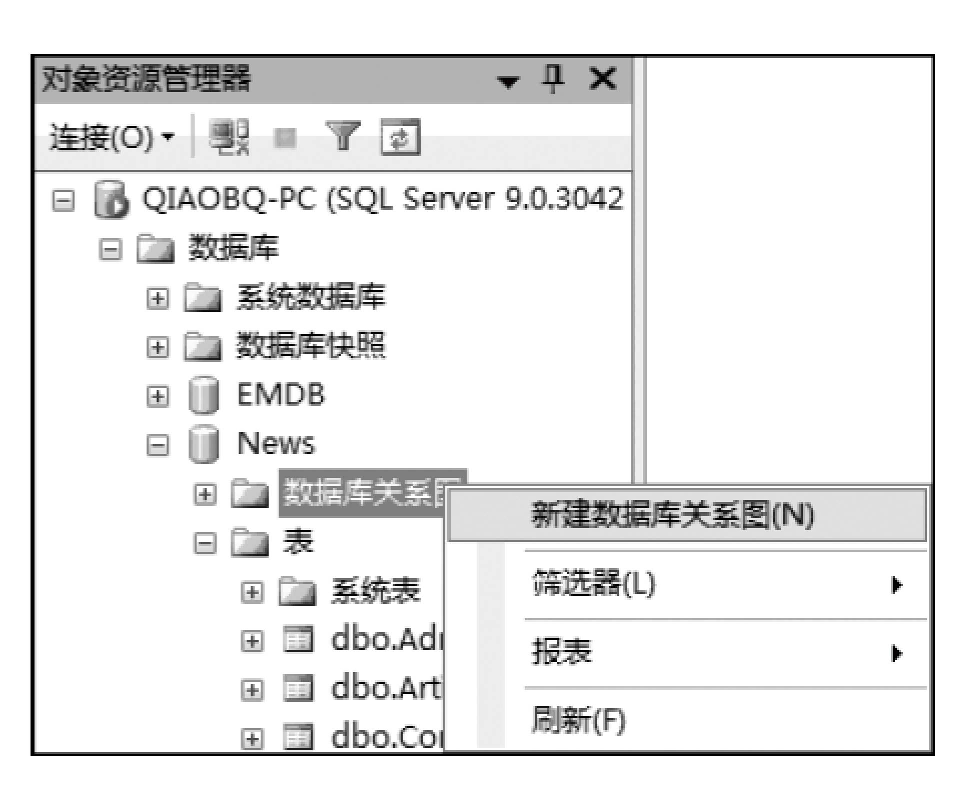


图 3-25　【数据库关系图】的快捷菜单

② 在弹出的如图 3-26 所示的【添加表】对话框中将 News 数据库中的表全部添加到关系图中，拖动表的标题调整好界面，根据图 3-2 和表 3-1～表 3-7 所示的内容设置主外键。

③ 选中需设置外键的表。以 Article 表为例，右击该表，出现快捷菜单，如图 3-27 所示，选择【关系】命令。



图 3-26　添加表



图 3-27　选择【关系】命令

④ 在弹出的【外键关系】对话框中单击【添加】按钮，如图 3-28 所示。

⑤ 打开【表和列】对话框，进行主键表和主键列、外键表和外键列的设置，如图 3-29 所示，设置完成后单击【确定】按钮。





图 3-28 外键关系设置

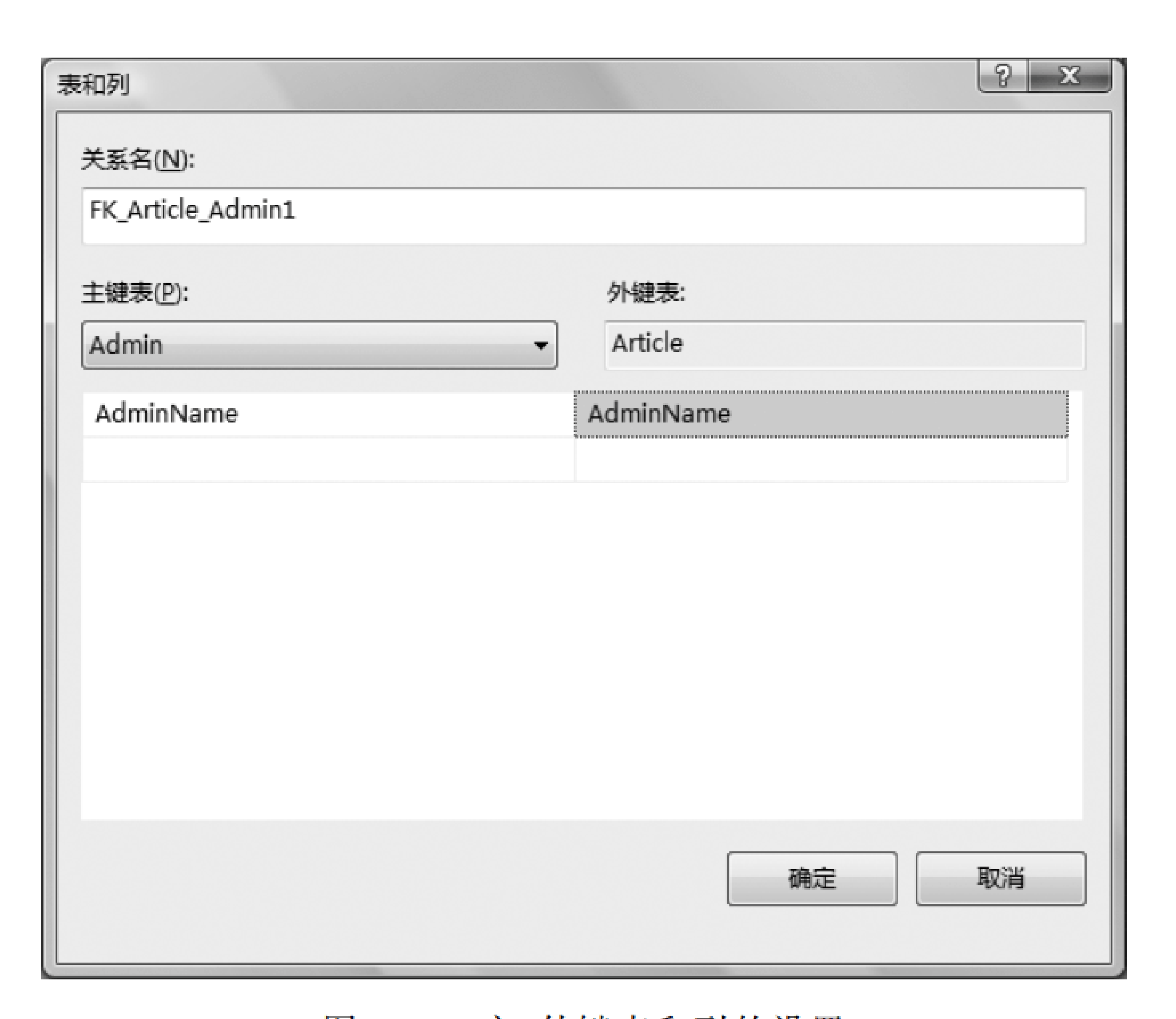


图 3-29 主、外键表和列的设置

⑥ 重复图 3-28 和图 3-29 所示方法，完成 Article 表的其他外键关系设置。

⑦ 重复图 3-27～图 3-29 所示方法，完成其他表的外键关系设置。

所有的表间关系设置完成后的关系图如图 3-30 所示，其中表间的连线代表关系，连线的钥匙端代表主键所在的表。



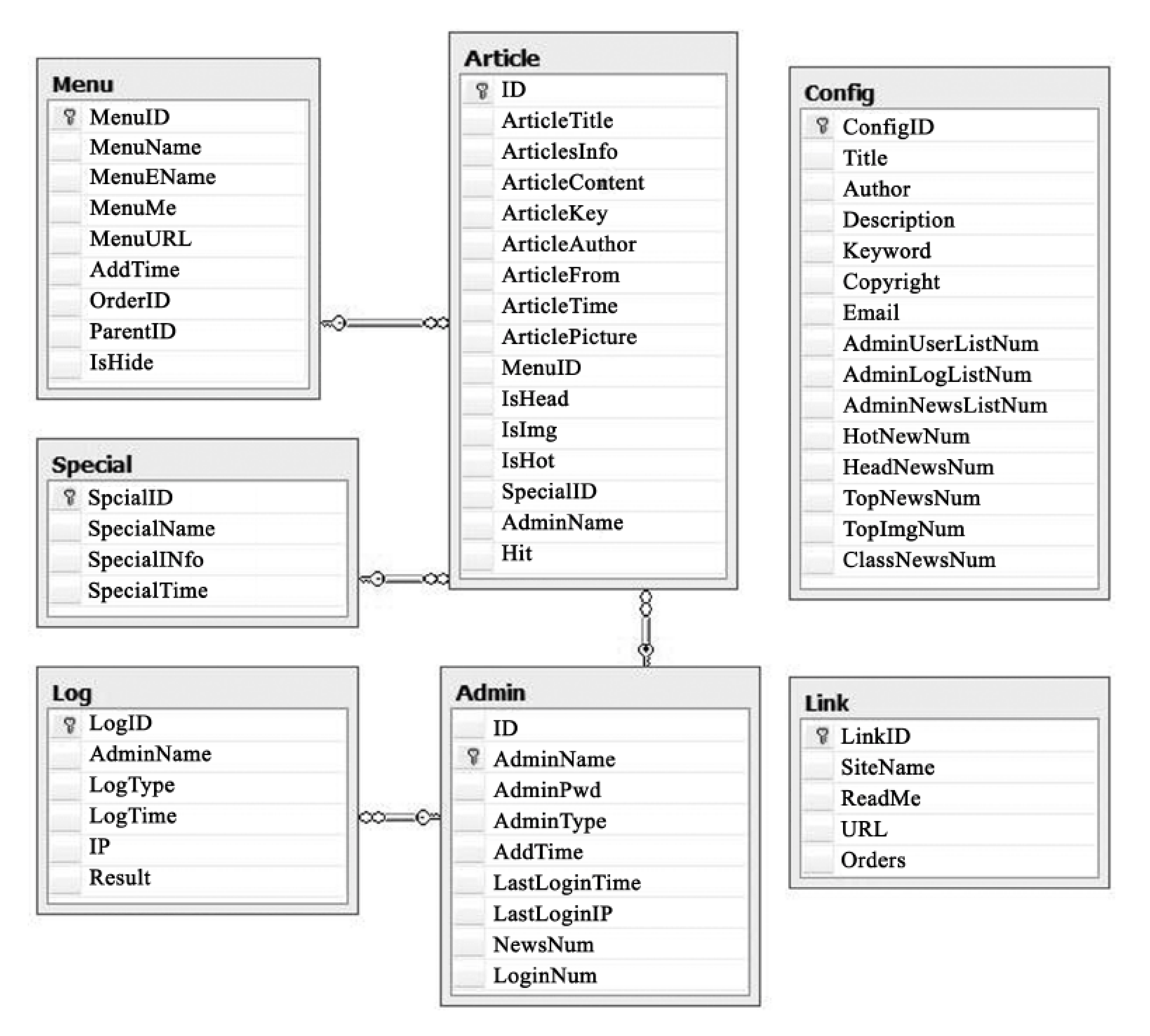


图 3-30　News 数据库的关系图

3.3.3　相关知识拓展

1. SQL Server 2005 中的数据类型说明

(1) 字符数据类型

字符数据是由任何字母、符号和数字任意组合而成的数据。字符数据类型分为非 Unicode 数据类型和 Unicode 数据类型。

① 非 Unicode 数据类型包括 char、varchar 和 text。

- char [(n)]：固定长度，非 Unicode 字符数据，长度为 n 个字符。n 的取值范围为 1～8000(含 8000)，存储大小是 n 个字节。
- varchar [(n | max)]：可变长度，非 Unicode 字符数据。n 的取值范围为 1～8000 (含 8000)。max 指示最大存储大小是 $2^{31}-1$ 个字节。存储大小是所输入字符的实际长度加 2 个字节。所输入数据的长度可以为 0 个字符。
- text：长度可变的非 Unicode 数据，最大长度为 $2^{31}-1$ (即 2147483647)个字符。

② Unicode 数据类型包括 nchar、nvarchar 和 ntext。

- nchar [(n)]：固定长度，Unicode 字符数据，长度为 n 个字符。n 的取值必须在 1～4000(含 4000)，存储大小为 2n 字节。
- nvarchar [(n|max)]：可变长度，Unicode 字符数据。n 的取值在 1～4000(含 4000)。max 指示最大存储大小为 $2^{31}-1$ 字节。存储大小是所输入字符个数的两



倍加上 2 个字节。所输入数据的长度可以为 0 个字符。

- ntext：长度可变的 Unicode 数据，最大长度为 $2^{31}-1$(即 1073741823)个字符。存储大小是所输入字符个数的两倍(以字节为单位)。

在 SQL Server 中，传统的非 Unicode 数据类型允许使用由特定字符集定义的字符。在 SQL Server 安装过程中，允许选择一种字符集。使用 Unicode 数据类型，列中可以存储任何由 Unicode 标准定义的字符。在 Unicode 标准中，包括了以各种字符集定义的全部字符。使用 Unicode 数据类型，所占用的空间是使用非 Unicode 数据类型所占用的空间大小的两倍。

例如大写字母 A 在 char 或者 varchar 数据类型的字段中，占用一个字节的存储空间，但是在 nchar 或者 nvarchar 数据类型的字段中占用两个字节。汉字"国"在这些数据类型的字段中都占用 2 个字节的存储空间。

在 SQL Server 中，Unicode 数据以 nchar、nvarchar 和 ntext 数据类型存储。使用这种字符类型存储的列可以存储多个字符集中的字符。当列的长度变化时，应该使用 nvarchar 字符类型，这时最多可以存储 4000 个字符；当列的长度固定不变时，应该使用 nchar 字符类型，同样，这时最多可以存储 4000 个字符；当使用 ntext 数据类型时，该列可以存储多于 4000 个字符。

(2) 数字数据类型

数字数据只包含数字。数字数据类型包括正数和负数、小数(浮点数)和整数。

整数由正整数和负整数组成，例如 39、25、－2 和 33967。在 SQL Server 中，整数存储的数据类型是 int、smallint 和 tinyint。

int 数据类型存储数据的范围大于 smallint 数据类型存储数据的范围。使用 int 数据类型存储数据的范围是从－2147483648 到 2147483647(每一个值要求 4 个字节存储空间)。

smallint 数据类型存储数据的范围大于 tinyint 数据类型存储数据的范围。使用 smallint 数据类型时，存储数据的范围从－32768～32767(每一个值要求 2 个字节存储空间)。

使用 tinyint 数据类型时，存储数据的范围是从 0～255(每一个值要求 1 个字节存储空间)。

decimal 和 numeric 是带固定精度和小数位数的数值数据类型。这种数据所占的存储空间根据该数据的小数点后的位数(即精度)来确定。numeric 在功能上等价于 decimal。

在 SQL Server 中，近似小数数据的数据类型是 float 和 real。float 和 real 是表示浮点数值数据的大致数值数据类型。由于浮点数据为近似值，因此，并非数据类型范围内的所有值都能精确地表示。

Bit 由 1 或者 0 组成。当表示真或者假、ON 或者 OFF 时，使用 Bit 数据类型。例如，"是否党员"可以存储在这种数据类型的列中。

(3) 日期和时间类型

日期和时间数据类型包括 datetime 和 smalldatetime 两种类型。

日期和时间数据类型由有效的日期和时间组成。例如，有效的日期和时间数据包括"4/01/98 12:15:00:00:00 pm"和"1:28:29:15:01 am 8/17/98"。前一个数据类型是日期在前，时间在后；后一个数据类型是时间在前，日期在后。在 SQL Server 中，日期和时间数据类型包括 datetime 和 smalldatetime 两种，datetime 所存储的日期范围是从 1753 年 1 月1 日开始，到

9999 年 12 月 31 日结束(每一个值要求 8 个存储字节)。smalldatetime 所存储的日期范围是 1900 年 1 月 1 日开始,到 2079 年 12 月 31 日结束(每一个值要求 4 个存储字节)。

2. 重命名表

在表创建完成以后,若需要修改表的名称,可以在该表名上右击,如图 3-31 所示,在弹出的快捷菜单中选择【重命名】命令,输入新的表名称,按回车即可。

3. 删除表

若要删除不需要的表,可以在图 3-31 中选择【删除】命令,在弹出的【删除对象】对话框中单击【确定】按钮,即可删除该表。



图 3-31　重命名表

3.3.4　经验总结

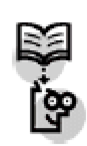
在表的结构设置中,数据类型为字符的列,其列宽的设置需要注意一下,对于每条记录来说,同一列在输入数据时不一定恰好输入相同的长度,比如 NewsTitle 列,其数据类型为 nvarchar(255),用于存储新闻标题。大家都清楚,不同的新闻其新闻标题未必是相同的,有的长一些,有的短一些,那该如何设置此列的列宽呢?答案是按最长的新闻标题取值宽度进行设置,也就是说,有的新闻标题长度是 50 字符,有的是 80 字符,但若据估计可能最长的新闻标题不会超过 255 个字符,此时便设置 NewsTitle 列的列宽是 255。

对于取值是由数字组成却没有数学意义或者以后不会进行数学计算的列(比如手机号码,由 11 位的阿拉伯数字组成,但其数据没有数学意义,人们通常不会说谁的手机号码比谁的手机号码大或小,也不会用手机号码进行数学运算),通常不设其数据类型为数值型,而是设为字符型。

列名的设置要注意,不要采取过于简单却毫无意义的列名,要尽可能使用能“顾名思义”的列名,不要怕长,目的是清晰,若是列名由多个英文字母组成,可以使用每个单词的首字母大写,其他字母小写的方式,使列名更加清晰明了,当然,表的命名及其他的数据库对象命名亦理当如此。

数据库中的对象还有许多,在“新闻类别网站”项目中只用到了表和数据库关系图这两个对象,在更加复杂的网站建设中,会用到其他的数据库对象,如存储过程、触发器和视图等,这里限于难度与篇幅,不再讲述。请有兴趣的读者继续参考其他数据库书籍或 SQL Server 2005 的帮助文档。

3.4　分离与附加数据库


知识储备

SQL Server 2005 中提供了分离数据库和附加数据库的功能。分离数据库功能将数据



库从 SQL Server 实例中删除，但并不删除数据库的数据文件和事务日志文件；附加数据库功能将分离后的物理文件重新附加到其他 SQL Server 实例中。利用分离和附加的功能可以实现数据库的复制或移动。

3.4.1 任务描述

将 News 数据库分离，复制到相应的计算机后，再实现该数据库的附加。

3.4.2 任务实现

本阶段的任务主要包括两个任务：

任务 1　分离 News 数据库

任务 2　附加 News 数据库

任务 1：分离 News 数据库

使用 SQL Server Management Studio 分离 News 数据库的具体操作步骤如下：

（1）在【对象资源管理器】窗口中，右击要分离的数据库 News，在弹出的快捷菜单中选择【任务】|【分离】命令，如图 3-32 所示。

（2）在【分离数据库】对话框中单击【确定】按钮，完成数据库的分离操作。分离成功后，可以发现【对象资源管理器】中已经没有 News 数据库了。

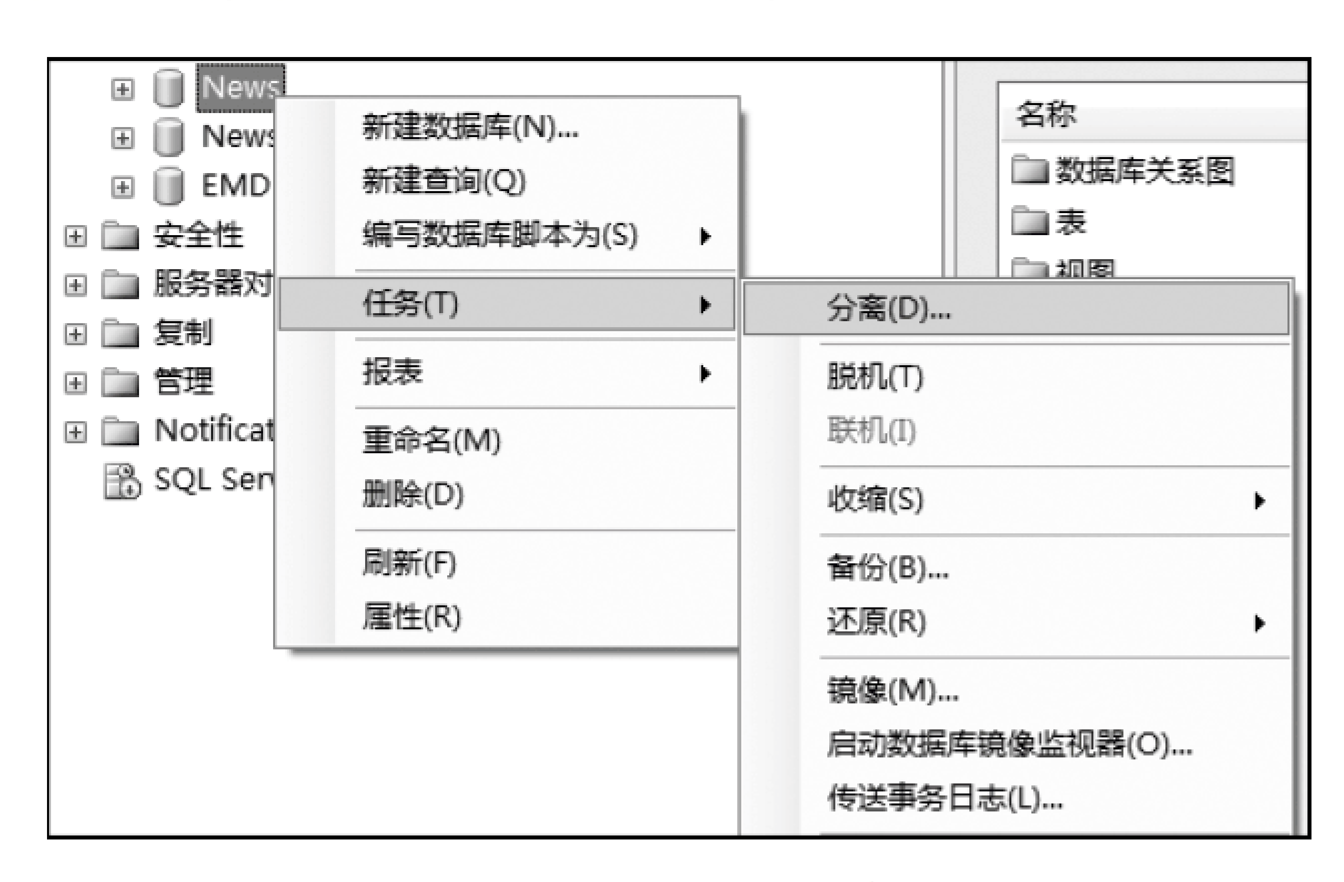


图 3-32　分离数据库

此时可以对数据库的物理文件进行复制或剪切，实现数据库物理文件的复制或移动。完成此操作后，接着做下面的任务 2。

任务 2：附加 News 数据库

使用 SQL Server Management Studio 附加 News 数据库的具体操作步骤如下：

（1）在【对象资源管理器】窗口中，右击【数据库】节点，在弹出的快捷菜单中选择【附加】命令。

（2）出现【附加数据库】对话框，单击【添加】按钮。

（3）出现【定位数据库文件】对话框，找到要附加的数据库文件，然后单击【确定】按钮。



(4) 回到【附加数据库】对话框，此时【News】数据库的内容已在图中详细地显示出来，如图 3-33 所示。修改【要附加的数据库】中的【附加为】列，即可为附加后的数据库重命名。

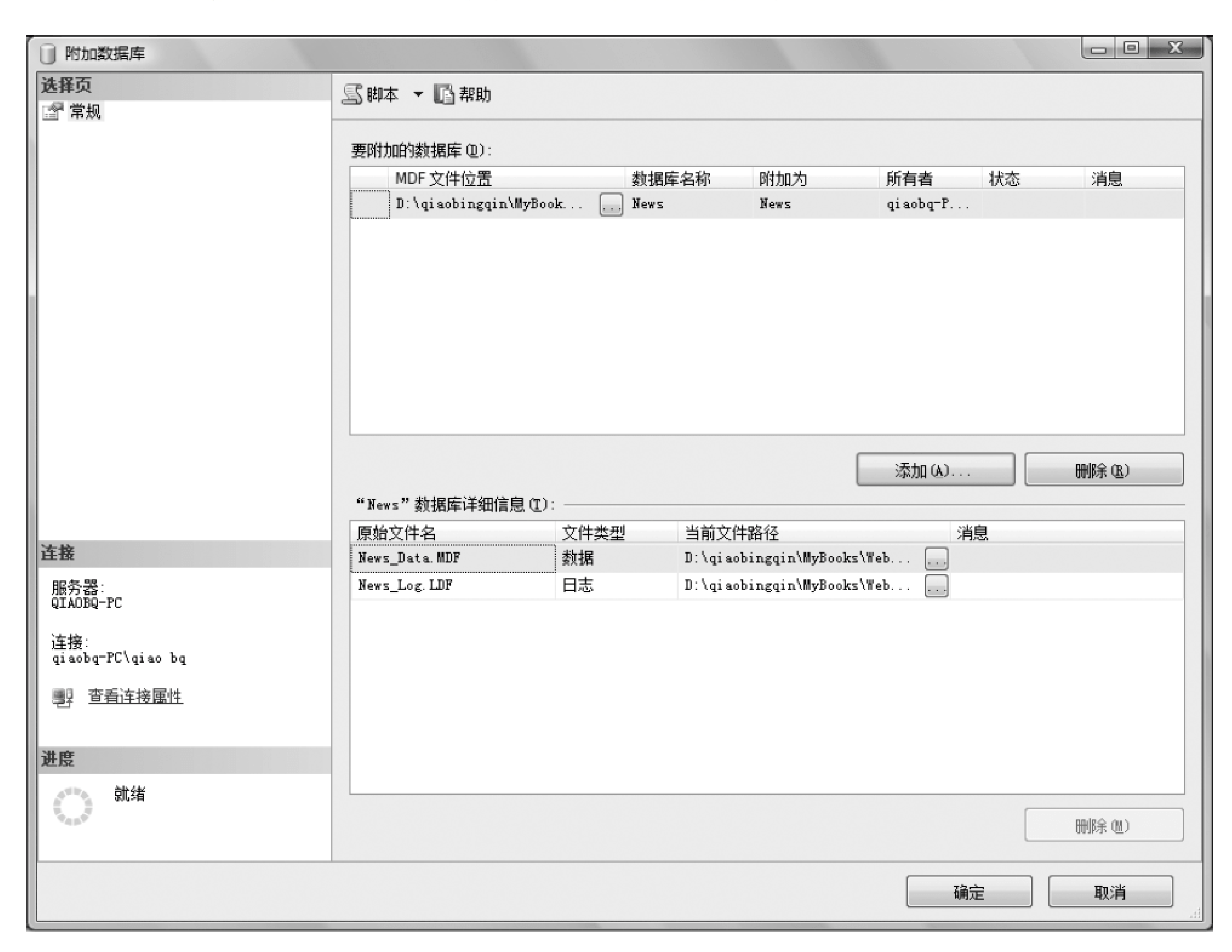


图 3-33　附加数据库

(5) 单击【确定】按钮，完成数据库附加操作。在【对象资源管理器】面板的【数据库】节点下即可看到该数据库。

3.4.3　相关知识拓展

1. 分离数据库

分离数据库能将数据库从 SQL Server 实例中删除，并使数据库在其数据文件和事务日志文件中保持不变。之后，可以使用这些文件将数据库附加到任何 SQL Server 实例(包括分离该数据库的服务器)。

在下面这些情况下数据库不能分离：

(1) 已复制并发布数据库。如果进行复制，则数据库必须是未发布的，必须通过运行 sp_replicationdboption 禁用发布后，才能分离数据库。

(2) 数据库中存在数据库快照。必须首先删除所有数据库快照，然后才能分离数据库。

(3) 数据库处于可疑状态。在 SQL Server 2005 中，无法分离可疑数据库，必须将数据库置入紧急模式，才能对其进行分离。

(4) 该数据库正在某个数据库镜像会话中进行镜像。除非终止该会话，否则无法分离



该数据库。

(5) 该数据库是系统数据库。

2. 附加数据库

在附加数据库时,所有的文件(mdf 和 ndf)必须是有效的,如果存储数据文件的路径发生变化,则必须指明当前数据库文件的路径。

附加数据库日志文件的要求在某些方面取决于数据库是“读—写”的还是“只读”的,下面分别进行说明:

(1) 对于“读—写”数据库,通常可以附加新位置中的日志文件。

(2) 如果附加的主数据文件是只读的,则数据库引擎假定数据库也是只读的。对于只读数据库,日志文件在数据库主文件中指定的位置上必须可用。因为 SQL Server 无法更新主文件中存储的日志位置,所以无法创建新的日志文件。即只读的数据库不能附加更换存储位置的日志文件。

3.4.4 经验总结

分离和附加数据库的操作对网站的发布是有影响的。当网站发布时,相应的数据库也可能因为需要,要从一台计算机复制或移动至另一台计算机,即对数据库物理文件进行复制或移动。此时必须先分离数据库,才能实现数据库的复制或移动。若没有分离数据库即对数据库物理文件进行复制或移动,在粘贴时会产生如图 3-34 所示的错误。

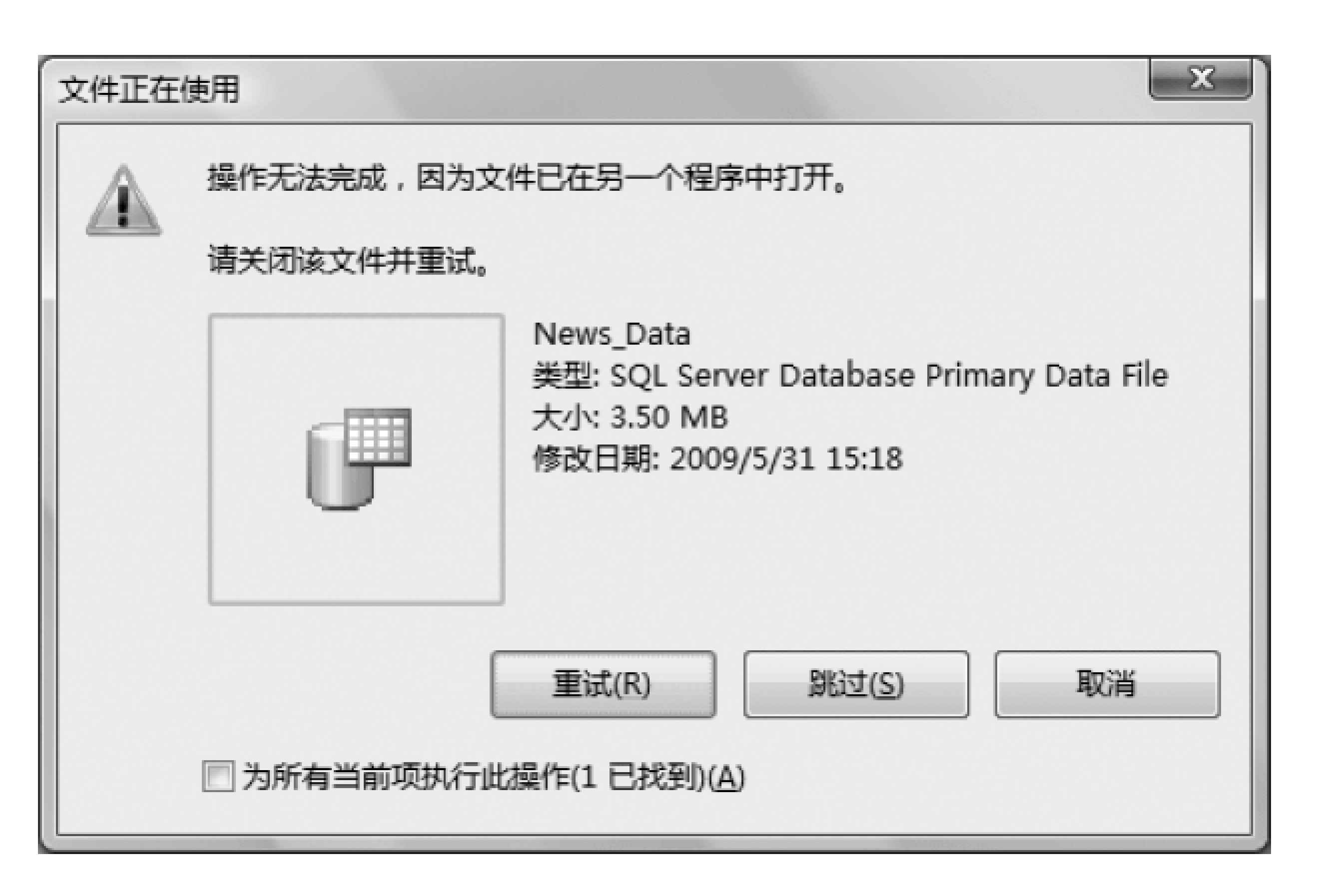


图 3-34 复制或移动数据库文件时的出错提示

解决该错误的方法是先将数据库进行分离,再进行复制或移动。成功地完成复制或移动后,再把新存储位置上的数据库文件进行附加,才可以操作该数据库。

3.5 实　训

按照本章的内容,完成“新闻类别网站”中的数据库创建、数据库中所有表的创建以及表间关系的创建。



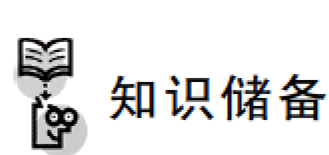


第4章　实现网站浏览器端的界面

网站浏览器端界面，就是用户所能看到的界面，是整个系统的表现层。

在第2章中进行了界面的设计，本章主要讲解针对设计图的具体实现过程，主要涉及网页制作工具 Adobe Dreamweaver CS3 的使用。此处重点讲解的是静态网页的制作，而且采用了较为先进的“DIV+CSS”布局，用到的主要知识是 XHTML 和 CSS，以静态网页的制作流程为主线依次讲解了站点的建立、站点的布局和规划、页面结构的规划与制作和具体页面的制作。

4.1　建立 Dreamweaver 站点



知识储备

Web 站点(网站)是一组具有共享属性(如相关主题、类似的设计或共同目的)的链接文档和资源。Dreamweaver 是一个站点创建和管理的工具软件，使用它不仅可以创建单独的文档，还可以创建完整的 Web 站点。为了达到最佳效果，在创建任何 Web 站点页面之前，应对站点的结构进行设计和规划，决定要创建多少页、每页上显示什么内容、页面布局的外观以及页是如何互相连接起来的。网站设计者可以在本地计算机上创建网页、跟踪和维护链接、管理网页和各种共享资源文件并将其上传到 Web 服务器，也可以随时在保存文件后传输更新的文件来对站点进行维护。

4.1.1　任务描述

建立“新闻类别网站”站点。

4.1.2　任务实现

具体操作步骤如下：

(1) 在本地硬盘上创建文件夹，并在其中创建一些子文件夹，用于存储站点中的网页和其他资源文件(如图像文件、Flash 动画等)。

(2) 通过【开始】|【所有程序】| Adobe Dreamweaver CS3 | Adobe Dreamweaver CS3 命令来启动 Dreamweaver。



小贴士

也可以通过桌面上的 Adobe Dreamweaver CS3 图标来启动。第一次启动 Dreamweaver 将设置默认编辑器,可以将指定文件的默认编辑器设置为 Adobe Dreamweaver CS3。

(3) 在 Dreamweaver 欢迎屏幕上单击【Dreamweaver 站点】,如图 4-1 所示。或者在菜单中选择【站点】|【新建站点】命令。



图 4-1　Dreamweaver 运行界面

(4) 当出现站点定义对话框时,选择【基本】选项卡,并在【您打算为您的站点起什么名字?】文本框中输入站点的名称,此处站点的名称只是 Dreamweaver 中为站点提供的一个标识,可以为任意字符;在【您的站点的 HTTP 地址(URL)是什么?】文本框中采用默认值,然后单击【下一步】按钮,如图 4-2 所示。

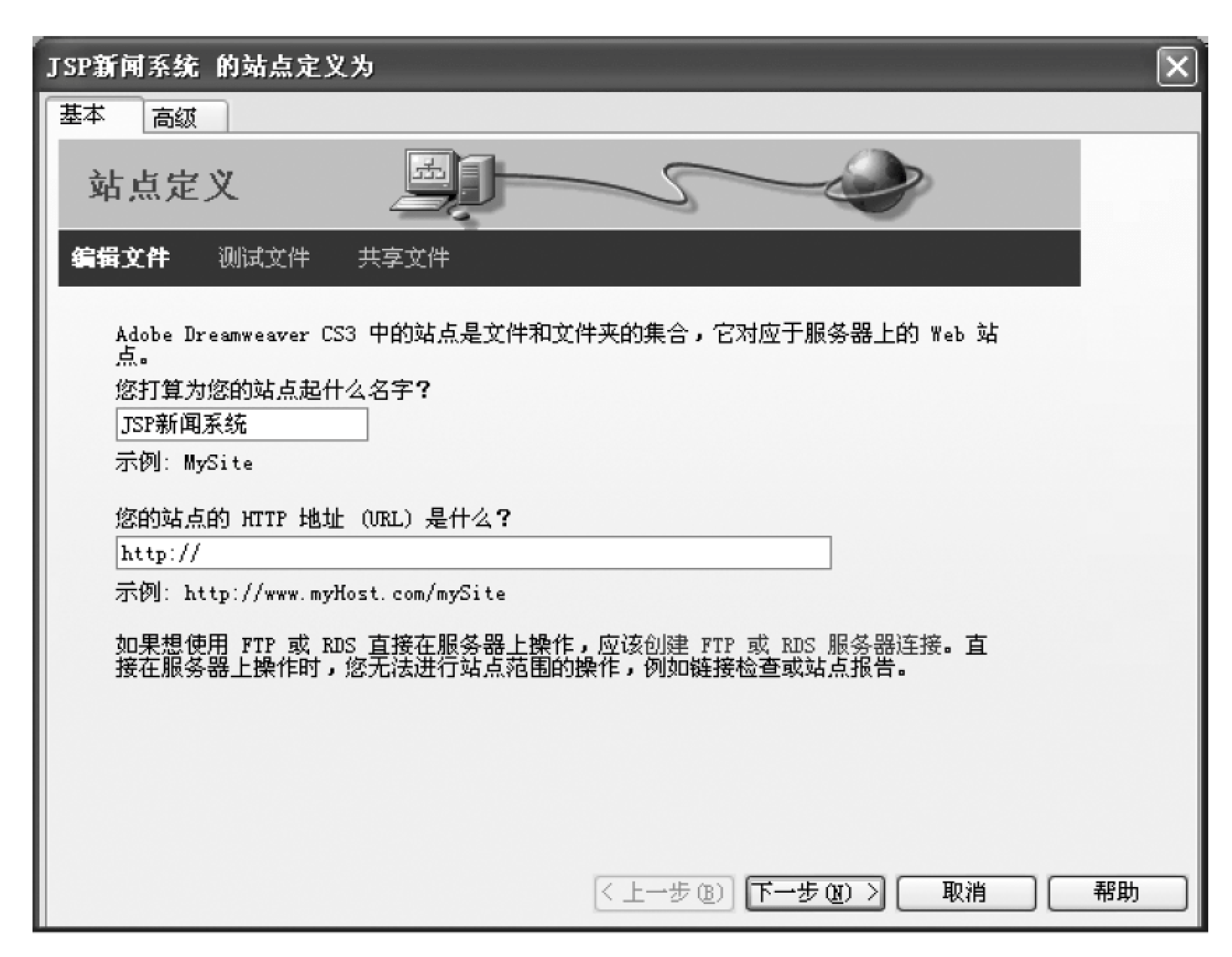


图 4-2　Dreamweaver 站点定义对话框

(5) 因为制作的是静态网页,不涉及后台服务器技术,所以在如图 4-3 所示的界面中选取【否,我不想使用服务器技术】单选按钮,然后单击【下一步】按钮。







图 4-3　是否使用服务器对话框

（6）在如图 4-4 所示的画面中，设置站点文件的存储位置。选择【编辑我的计算机上的本地副本，完成后再上传到服务器(推荐)】单选按钮，并在【您将把文件存储在计算机上的什么位置?】文本框中输入站点所在的根目录的路径，或者通过单击文件夹图标来选择存储路径，然后单击【下一步】按钮。如果站点在远程服务器上而不在本地，也可以选择【使用本地网络直接在服务器上进行编辑】单选按钮。



图 4-4　设置站点文件的存储位置对话框



（7）在如图 4-5 所示的画面中，选择连接远程服务器的方式。在此进行的是前台模板的制作，是纯静态的网页，并不涉及远程服务器，所以在“您如何连接到远程服务器？”列表框中选择【无】选项，即完成设置，然后单击【下一步】按钮。



图 4-5　远程服务器的连接方式设置

（8）在如图 4-6 所示的画面中，列出了所建站点的汇总信息，认真核查站点的设置情况，核查无误后单击【完成】按钮，即可完成站点定义。



图 4-6　站点的汇总信息界面





（9）此时，将回到如图 4-7 所示的站点管理界面，单击【完成】按钮，即可完成站点建立，新建的站点出现在【文件】面板中，至此，Dreamweaver 站点的建立结束。



图 4-7　Dreamweaver 启动画面

小贴士

使用【文件】面板可以执行以下任务：访问站点、服务器和本地驱动器；查看文件和文件夹；管理文件和文件夹；使用站点的可视化地图。



图 4-8　Dreamweaver 启动画面

使用【文件】面板可以管理文件并在本地计算机和远程服务器之间传输文件。当在本地站点和远程站点之间传输文件时，会在这两种站点之间维持平行的文件和文件夹结构。如果站点中不存在相应的文件夹，则 Dreamweaver 将创建这些文件夹。也可以在本地和远程站点之间同步文件，Dreamweaver 会根据需要在两个方向上复制文件，并且在适当的情况下删除不需要的文件。

在【文件】面板中选中站点的根目录并右击，弹出快捷菜单，如图 4-8 所示，选择【新建文件】或【新建文件夹】命令，在站点下创建所需的文件或文件夹，选择其他命令便可进行相应的站点管理，在此不再详述。



4.1.3 相关知识拓展

一个完整的 Dreamweaver 站点应由三个部分组成，但是 Dreamweaver 站点的具体组成主要取决于开发环境和所开发网站的类型。

（1）本地根文件夹。可以位于本地计算机或网络服务器上，用于存储正在处理的文件，此文件夹也称为本地站点。如果要定义一个 Dreamweaver 站点，则至少要设置一个本地根文件夹。

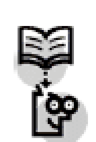
（2）远程文件夹。位于运行 Web 服务器的计算机上，存储用于测试、生产和协作等用途的文件，此文件夹也称为远程站点。如果已经申请了域名并且在 Internet 上拥有空间，则需要设 JR 远程文件夹。利用 Dreamweaver 站点管理功能可以在本地硬盘与 Web 服务器之间传输文件。

（3）测试服务器文件夹。Dreamweaver 在其中处理动态网页的文件夹。如果要运行动态网页并连接到数据库，则需要设置测试服务器文件夹。如果 Dreamweaver 和服务器软件运行在同一台本地计算机上，也可以把本地根文件夹和测试服务器文件夹设置为同一个文件夹。

4.1.4 经验总结

在 Dreamweaver 中建立站点，是进行网站开发的关键步骤。这里所讲的建立站点，其实就是在 Dreamweaver 中定义站点、策划站点结构、部署开发环境。定义站点是为了更好地利用【文件】面板对站点文件进行管理，也是为减少一些错误的出现，如路径出错、链接出错等。许多初学者开始做网页时，只把注意力放在制作单一网页上，对文件的条理性、结构性不加以管理，没有对文件进行分类管理，使整个站点结构显得很乱，所以一定要注意，在开发网站之前，认真策划好站点结构。

4.2 网站的整体布局和规划

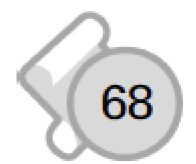


知识储备

1. CSS 盒子模型

在网页设计中常听到这样的属性名：内容（content）、填充（padding）、边框（border）、边界（margin），这些属性可以用日常生活中的盒子（箱子）来理解。日常生活中所见的盒子可以放置物品，怕物品被挤压、摔打造成损坏，所以要在盒子里填充些泡沫或者其他抗震的辅料，假设有多个这样的盒子，在摆放时需要留有一定的空隙保持通风，同时也为了方便取出盒子。CSS 的上述属性也可以这样理解，所以把 CSS 称为盒子模式。对于 CSS 来讲，其边框可以理解为盒子本身，内容是盒子里装的东西，填充是添加的泡沫或者其他抗震的辅料，边界是盒子摆放时留的空隙。CSS 盒子示意如图 4-9 所示。

但 CSS 盒子与现实中的盒子也有所不同。在网页设计上，CSS 的内容属性常指文字、





图片等元素，但是也可以是小盒子(DIV 嵌套)。生活中的盒子放置物品时，物品一般不能大于盒子，否则盒子会被撑坏的，但 CSS 盒子具有“弹性”，放置的内容大过盒子本身时会把盒子撑大，但不会损坏盒子。填充属性只有宽度属性，就像生活中盒子里的抗震辅料厚度。边框属性有粗细和颜色之分，类似生活中盒子的厚度以及盒子的外观颜色。边界属性代表该盒子与其他元素间要保留多大距离。

基于 CSS 的网站设计就像把不同大小和颜色的盒子，以一定的间隙和顺序摆放好，从高处看这些盒子，它们所构成的图形结构就是要做的网页版面设计，如图 4-10 所示。

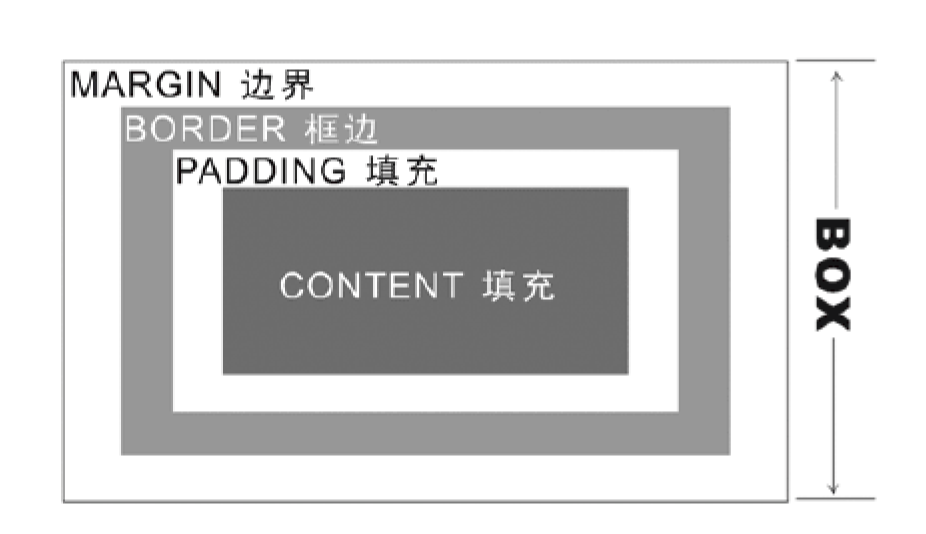


图 4-9　CSS 盒子模式

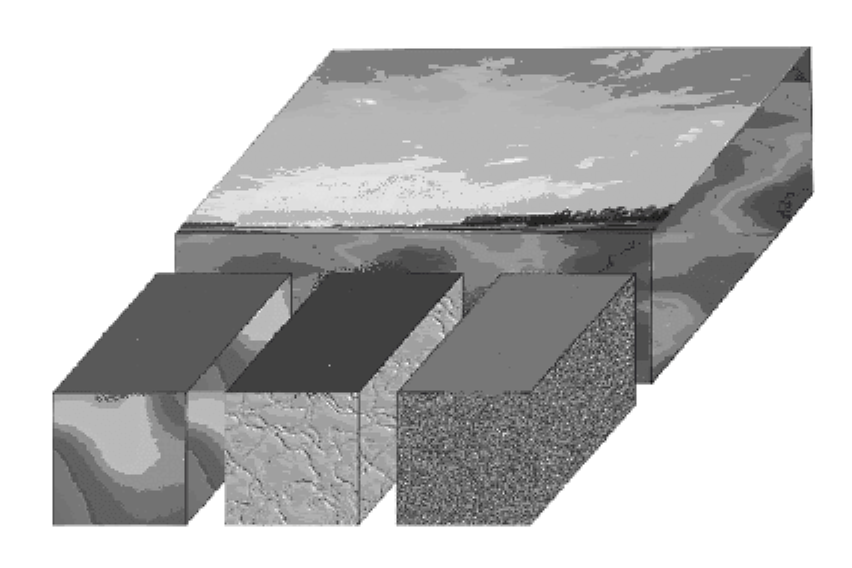


图 4-10　由“盒子”堆出来的网页版面

2. CSS 的设计思路

传统的前台网页设计是这样进行的：根据要求，先考虑好主色调，要用什么类型的图片，用什么字体、颜色等，然后再用 Photoshop 这类软件“自由”地画出设计结果，最后再切成小图，再“不自由”地通过设计 HTML 生成页面。

改用 CSS 设计网页版面后，要转变传统的设计思想。CSS 设计思路主要考虑的是页面内容的语义和结构，一个强 CSS 控制的网页，在完成网页制作后，还可以轻松地改变想要的网页风格。使用 CSS 设计版面的另外一个目的是让代码易读，区块分明，强化代码重用。

3. 用 CSS 实现结构与表现分离

用 CSS 布局网页可以实现页面结构和表现形式相分离，结构与表现分离，网页代码才简洁，更新网页时才方便。

4. 使用 CSS 设计网页的步骤

(1) 用 DIV 定义语义结构，即像盖楼一样利用 DIV 搭建基本结构；

(2) 用 CSS 属性美化网页，如设置背景、线条边框、对齐属性、字体、颜色等；

(3) 在 CSS 定义的盒子内加上内容，如文字、图片等(没有表现属性的标签)。

小贴士

如果网页设计的很复杂，能不能实现上述的 CSS 的效果？

答案是肯定的。如果用 CSS 实现不了的效果，一般用表格(table，HTML 的元素)也是很难实现的，CSS 的控制能力比表格要强大得多。如果使用 CSS 排版网页，对于网页设计效果不满意的地方，特别是色调，改起来相当容易，甚至可以定制几种风格的 CSS 文件对网页进行对比选择，或者写一个程序实现动态调用，让网站具有动态改变风格的功能。

4.2.1　任务描述

规划“新闻类别网站”的整体版面布局。



4.2.2 任务实现

1. 分析设计图

分析设计图实际上就是依据网页属性的实际情况，将设计图进行划分，使设计的效果能完美地在网页中实现。实际上在第 2 章网站设计的过程中这种划分就已经体现出来了，这里考虑到设计和制作可能不是一个人，因此站在制作的角度对设计图进行了分析。此处分析三个主要页面：首页、列表页和内容页的设计图，可得出如图 4-11 所示的页面框架图，在此只是从整体上把握页面的框架结构，针对页面具体部分的细节分析将在后面有针对性地进行。

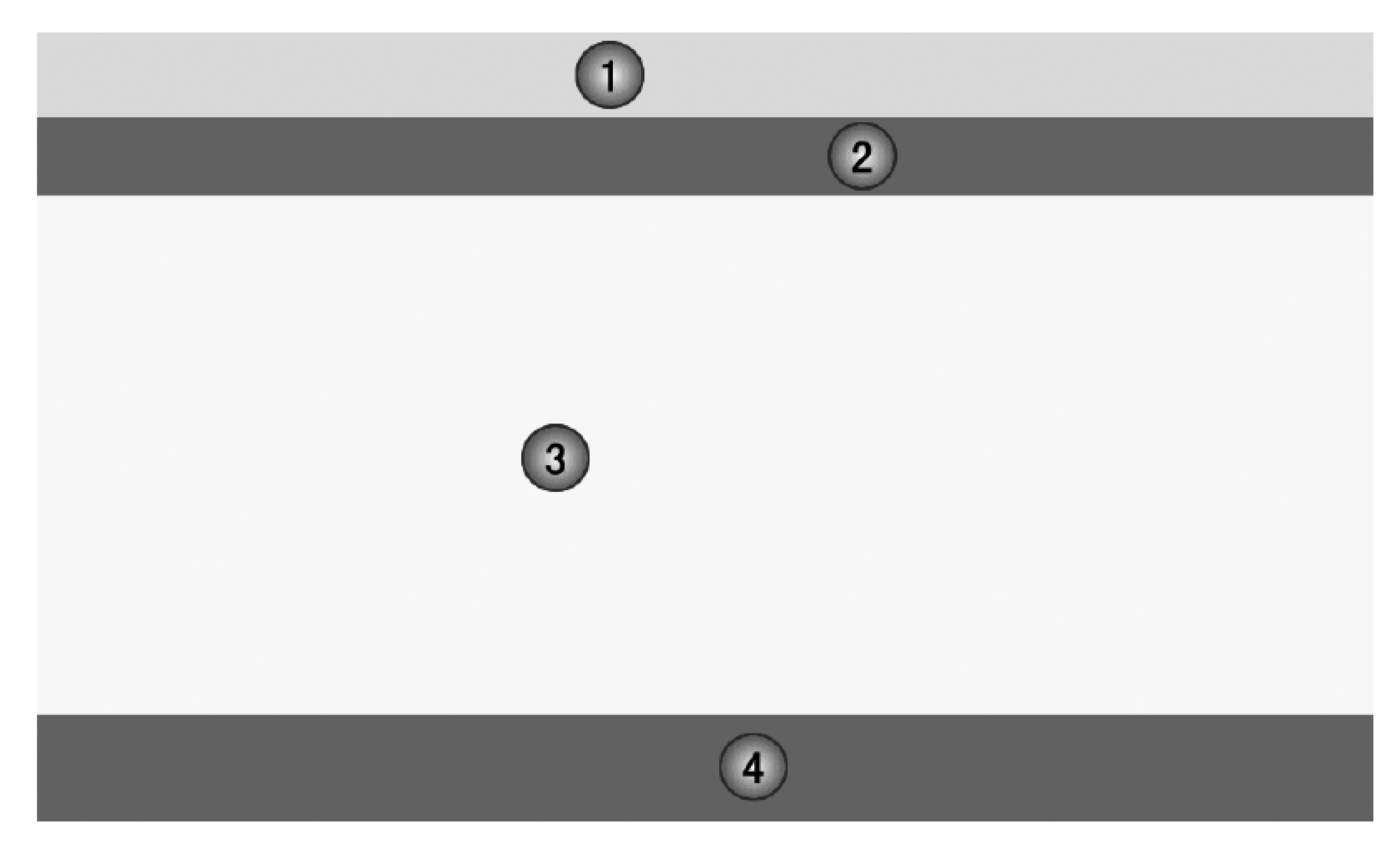


图 4-11　划分设计图的结果

2. 具体参数分析

通过上面的分析，可以得出一些构成页面的重要参数，具体如下：

① 网站头部图标（TOP），包含网站的 Logo 和站名。参数：Width 为 900px，Height 为60px。

② 导航条（Main Navigation），具有按钮特效。参数：Width 为 900px，Height 为50px。

③ 网站的主要内容（Content ）。参数：Width 为 900px，Height 为 Changes，depending on content。

④ 网站底栏，包含版权信息等（Footer）。参数：Width 为 900px，Height 为 60px。

⑤ 背景。参数：采用纵向重复的背景图。

4.2.3 相关知识拓展

使用 DIV＋CSS 结构建立网页的优点有：

（1）可以大大缩减页面代码，提高页面浏览速度，缩减网络带宽成本。

（2）网页结构清晰，用只包含结构化内容的 HTML 代替嵌套的标签，搜索引擎能更有



效地搜索到相关内容。

(3) 实现网页内容与表现形式的分离，从而缩短网站改版时间。只要简单地修改几个 CSS 文件就可以重新让一个有成百上千个页面的站点换新颜。

(4) 具备强大的字体控制和排版能力。CSS 控制字体的能力比 FONT 标签好得多，有了 CSS，不再需要用 FONT 标签或者其他的元素来控制网页表现风格、改变字体颜色、字体样式等。

(5) CSS 非常容易编写，可以像写 HTML 代码一样轻松地编写 CSS。

(6) 使用<table>进行布局时，会产生很多垃圾代码，一些修饰的样式及布局的代码混合一起，很不直观。而 DIV 更能体现样式和结构相分离，结构的重构性强。

(7) 几乎所有的浏览器都可以支持 CSS。

(8) 以前一些必须通过图片转换实现的页面功能，现在只要用 CSS 就可以轻松实现，从而提高页面下载的速度。

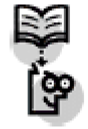
(9) 使用 CSS 可以轻松地控制页面的布局，使页面的字体变得更漂亮，更容易编排。

4.2.4　经验总结

对于设计图的分析，是个仁者见仁智者见智的事情，当看到一个设计效果时要想能很好地实现，最重要的是看实现人员对 HTML 和 CSS 样式属性的掌握程度，熟练地掌握这些技术是进行分析的前提，否则再好的设计图也没办法实现。这就好比有一个非常漂亮的大楼设计图，但是工人技术不行无法盖起来一样。

除了要熟练掌握 HTML 和 CSS 样式属性，还要多做网页、多看网页、多思考，在实践中掌握一些效果的制作原理。

4.3　网站页面结构的制作


知识储备

1. XHTML

XHTML(eXtensible HyperText Markup Language)，可扩展的超文本标记语言。国际 W3C 组织(World Wide Web Consortium)于 2000 年发布了 XHTML 1.0 版本，这是一种在 HTML 4.0 基础上优化和改进的新语言，是一种基于 XML 应用的 HTML，它的可扩展性和灵活性将适应未来 Web 应用的更多需求。

虽然 XML 数据转换能力强大，完全可以替代 HTML，但面对成千上万已有的基于 HTML 语言设计的网站，目前直接采用 XML 尚为时过早。因此，使用 XML 的规则对 HTML 4.0 进行扩展，便形成了 XHTML，建立 XHTML 的目的就是实现 HTML 向 XML 的过渡。目前国际上在网站设计中推崇的 Web 标准就是基于 XHTML 的应用。

简单地理解，XHTML 其实就是严谨而准确的 HTML。如果说 HTML 是汉语，那么 XHTML 就是标准普通话。如果是刚开始学习网页设计的话，最佳的选择是直接学习 XHTML。



XHTML 属于 HTML 家族，对比以前各个版本的 HTML，它具有更严格的书写标准、更好的跨平台能力。XHTML 将以前的版本 HTML 能够实现的一些功能交给了 CSS，这意味着学习网页制作将需要学习两种技术。

目前 HTML 正在向 XHTML 过渡发展，因此创建 HTML 网页时应该对 XHTML 代码规范有所了解，以便设计出符合标准的网页。XHTML 代码规范主要包括以下规则。

(1) 所有标记必须使用相应的结束标记进行关闭

在 HTML 中，成对的标记可以不用结束标记来关闭。但在 XHTML 中要求所有成对的标记必须使用结束标记来关闭，例如<head>…</head>，<title>…</title>以及<body>…</body>等；对于那些不成对的标记，则必须在其最后使用一个正斜线符号"/"来关闭，例如
，<hr />，<input…/>等。

(2) 标记及其属性名称必须使用小写字母

在 HTML 中是不区分字母的大小写的，例如<HTML>、<Html>或<html>作用都是一样的。在 XHTML 中，规定所有标记及其属性名称都必须使用小写字母。例如，正文标记必须写成<body></body>，而不允许写成<BODY></BODY>。

(3) 标记的属性值必须使用引号括起来

在 HTML 中，设置标记的属性值可以不使用引号，例如。在 XHTML 中，必须使用双引号把属性值括起来，例如>。

(4) 标记的属性必须具有值

在 HTML 中，设置标记的属性值可以使用简写形式，例如：<input checked>。在 XHTML 中，要求所有属性都必须具有值，而不允许使用简写形式，对于 HTML 中那些可以简写的属性，在 XHTML 中把属性名称设置为属性值就可以了，如<input checked="checked">。

(5) 强制 XHTML 元素

在 XHTML 规范中，要求所有文档都必须有一个<! DOCTYPE>声明，因此创建符合 XHTML 1.0 标准的网页，必须在文档首行添加以下文档类型声明：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

其中 DOCTYPE 是 Document Type(文档类型)的缩写，说明编码使用的 XHTML 或者 HTML 的版本，其中的 DTD(例如 xhtml1-transitional.dtd)称为文档类型定义，浏览器根据所定义的 DTD 来解释页面中的 HTML 标记并将其内容呈现出来。要创建符合标准的网页，<! DOCTYPE>声明是必不可少的。

2. CSS

CSS 是 Cascading Style Sheets(层叠样式表单)的简称。CSS 是一种标记性语言，与 HTML 是"表兄弟"，样式表技术诞生于 1996 年，需要浏览器支持，现在大多数浏览器都支持 CSS。更多的人把它称做样式表，顾名思义，它是一种设计网页样式的工具。借助 CSS 的强大功能，网页将在设计者丰富的想象力下千变万化。

使用 HTML 语言可以构建网页的文档结构，通过 HTML 标记可以定义各种页面元素



并通过属性控制页面元素的样式。为了增强或控制页面元素的样式并实现样式信息与页面内容的分离，引入了 CSS。CSS 样式类似于面向对象编程中的类，可以一次定义、重复使用，使用 CSS 有助于实现网站的统一布局。

CSS 可以精确定义页面中各元素以及页面的整体样式，可以非常灵活地控制页面的确切外观，控制许多文本属性，包括特定字体和字号，粗体、斜体、下画线、文本阴影、文本颜色、背景颜色、链接颜色和链接下画线等，还可以确保在多个浏览器中以更一致的方式处理页面布局和外观。

CSS 不仅可以控制一篇文档中的文本格式，而且还可以控制多篇文档的文本格式。因此使用 CSS 样式表定义页面文字，将会使工作量大大减小。

CSS 是一组格式设置规则，可以将所有关于文档的样式从内容中全部脱离出来，页面内容(即 HTML 代码)存放在 HTML 文件中，定义内容表示形式的 CSS 规则存放在另一个文件(外部样式表)或 HTML 文档的另一部分(通常为网页首部)中。

术语"层叠"表示对同一个页面元素应用多种样式的能力。例如，可以创建一个 CSS 规则来应用颜色，创建另一个 CSS 规则来应用边距，然后将两者应用于页面上的同一段文本。所定义的样式向下"层叠"到页面元素，并最终实现想要的设计效果。

3. CSS 的引用

可以用以下三种方式将样式表嵌入网页，最接近目标的样式定义优先权最高。高优先权样式将继承低优先权样式的未重叠定义但覆盖重叠的定义。

(1) 链入外部样式表文件(Linking to a Style Sheet)

先建立外部样式表文件(.css)，然后使用 HTML 的 link 对象。

【例 4-1】

```
<head>
  <title>文档标题</title>
  <link rel=stylesheet href="http://www.dhtmlet.com/dhtmlet.css" type="text/css">
</head>
```

(2) 定义内部样式块对象(Embedding a Style Block)

可以在 HTML 文档的<HTML>和<BODY>标记之间插入一个<STYLE>…</STYLE>块对象。

【例 4-2】

```
<html>
<head>
    <title>文档标题</title>
    <style type="text/css">
    <!--
    body {font: 10pt "Arial"}
    h1 {font: 15pt/17pt "Arial"; font-weight: bold; color: maroon}
    h2 {font: 13pt/15pt "Arial"; font-weight: bold; color: blue}
    p {font: 10pt/12pt "Arial"; color: black}
    -->
    </style>
  </head>
```



```
<body>
```

注意

这里将 style 对象的 type 属性设置为 text/css，是允许不支持这种类型定义的浏览器忽略样式表单。

(3) 内联定义(Inline Styles)

内联定义是指在对象的标记内使用对象的 style 属性定义适合该对象的样式表属性。

【例 4-3】

```
<P style="font: '黑体';font-size: 24px;color: #FF0000;">我的家乡是山西</P>
```

除设置文本格式外，还可以使用 CSS 控制网页中块级元素的格式和定位。块级元素是一段独立的内容，在 HTML 中通常由一个新行分隔，并在视觉上设置为块的格式。例如，H1 标记、P 标记和 DIV 标记都会在网页上生成块级元素。对块级元素可以执行以下操作：为它们设置边距和边框，将它们放置在特定位置，向它们添加背景颜色，在它们周围设置浮动文本等。对块级元素进行操作的方法实际上就是使用 CSS 进行页面布局设置的方法。

4. CSS 语法

CSS 样式表由一些规则组成，每个规则由选择符和属性声明两个部分组成，其中选择符用于标识格式元素(如例 4-2 中的 p、h1、h2，类名或 id 都是选择符)，属性声明则用于定义元素的样式。属性声明用花括号括起来，其内容由一些属性—值对组成，属性名称与属性值用冒号(：)分隔，不同属性—值对用分号(；)分隔。样式规则的基本语法为：

```
选择符{属性: 值}
```

或者

```
选择符{属性 1: 值 1;属性 2: 值 2}
```

5. CSS 的类型

CSS 的类型有三种。

(1) 重定义 HTML 标签：为 HTML 标识符重新定义格式化。

(2) 自定义样式：可以将自定义的样式应用于任何文本。

(3) 使用 CSS 选择器：专用于控制链接的显示效果，共有四种选择器。

- a：link(没有接触过的链接)：用于定义链接的常规状态。这种链接状态是基本的，推荐至少要定义这种链接。
- a：hover(鼠标放在链接上的状态)：用于产生视觉效果。
- a：visited(访问过的链接)：用于已经访问过的链接。颜色要和普通文本的链接和其他状态颜色不同。
- a：active(在链接上按下鼠标时的状态)：用于表现鼠标按下时的链接状态。

以上选择器中，第 4 种较少使用。在定义选择器时，必须严格按此顺序定义，否则可能会无效。



4.3.1 任务描述

利用 Adobe Dreamweaver CS3 进行"新闻类别网站"页面结构的制作，效果如图 4-12 所示。



图 4-12　"新闻类别网站"页面结构效果图

4.3.2 任务实现

该任务可具体细分为基本框架的制作、页面头部的制作、页脚部分的制作和导航条的制作，所以通过以下四个任务来实现。

任务 1：基本框架的制作
任务 2：页面头部的制作
任务 3：页脚部分的制作
任务 4：导航条的制作

任务 1：基本框架的制作

此任务主要是建立网站基本的框架，完善字体集、字体大小、页面的背景、超链接和段落等整体效果的控制。由设计图分析可知，背景效果采用了图片的纵向重复来制作。具体效果如图 4-13 所示。

具体操作步骤如下：

(1) 打开 4.1 节所建的站点。一般情况下启动 Dreamweaver 后，自动打开的就是上次使用过的站点。如果有多个站点，可以在【文件】面板中进行选择，如图 4-14 所示。

(2) 新建标准的 XHTML 页面。通过菜单栏的【文件】|【新建】命令，打开【新建文档】对话框，依次选择【空白页】| HTML |<无>选项，从"文档类型"列表框中选择 XHTML 1.0 Transitional 选项，然后单击【创建】按钮，如图 4-15 所示。



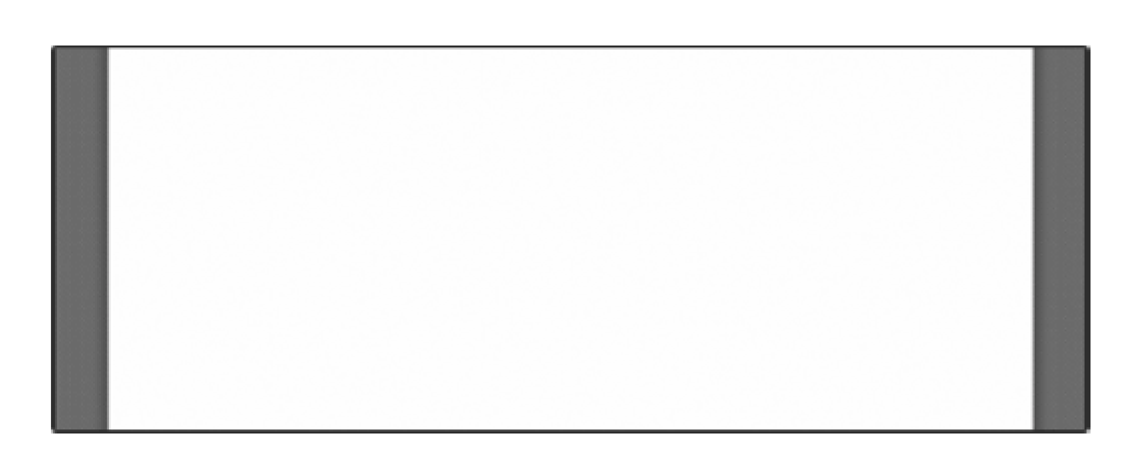


图 4-13 “新闻类别网站”基本框架的效果图



图 4-14 在【文件】面板中进行站点的切换



图 4-15 【新建文档】对话框

(3) 保存刚刚新建的 HTML 文件。通过菜单栏的【文件】|【另存为】命令，打开【另存为…】对话框，在文件名处输入“Index.html”，其他参数默认，然后单击【创建】按钮，如图 4-16 所示。

(4) 建立 DIV 结构。按照如程序 4-1 所示的代码完善 Index.html。

【程序 4-1】 建立 DIV 结构

Example Source Code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;charset=gb2312" />
<title>JSP新闻系统</title>
<meta http-equiv="Content-Language" content="en-us" />
<meta http-equiv="imagetoolbar" content="no" />
```





图 4-16　【另存为…】对话框

```
<meta name="MSSmartTagsPreventParsing" content="true" />
<meta name="description" content="JSP 新闻系统" />
<meta name="keywords" content="JSP,新闻系统" />
<meta name="author" content="Skimob" />
</head>

<body>
<div id=Top></div>
<div id=MenuBar></div>
<div id=Main></div>
<div id=Footer></div>
</body>
</html>
```

(5) 按照上述操作步骤建立新文件，名为 Style.css 的 CSS 文件，保存到 Themes 目录下，并按照如程序 4-2 所示的代码建立相应的 CSS 结构。

【程序 4-2】　建立相应 CSS 结构

Example Source Code:

```
@charset "utf-8";
/* CSS Document */
 * {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    margin: 0px auto;
    padding: 0px;
    text-decoration: none;
}
a{
    color: #990000;
```



```
}
a: hover {
    color: #FF0000;
}
a: visited {
    color: #990033;
}
body {
    font-size: 1.5ex;
    line-height: 180%;
    background-color: #CC0000;
    background-image: url(Images/bg.jpg);
    background-repeat: repeat-y;
    background-position: center;
}
P {
    margin-right: 10px;
    margin-left: 10px;
    text-align: left;
    text-indent: 2em;
}
```

注意

此处对背景的设置，涉及对设计图的分析，应特别注意。

(6) 建立 HTML 和 CSS 文件的连接。如图 4-17 所示，在【属性】面板的【样式】下拉列表中选择“附加样式表…”选项，弹出【链接外部样式表】对话框，在对话框中单击【浏览】按

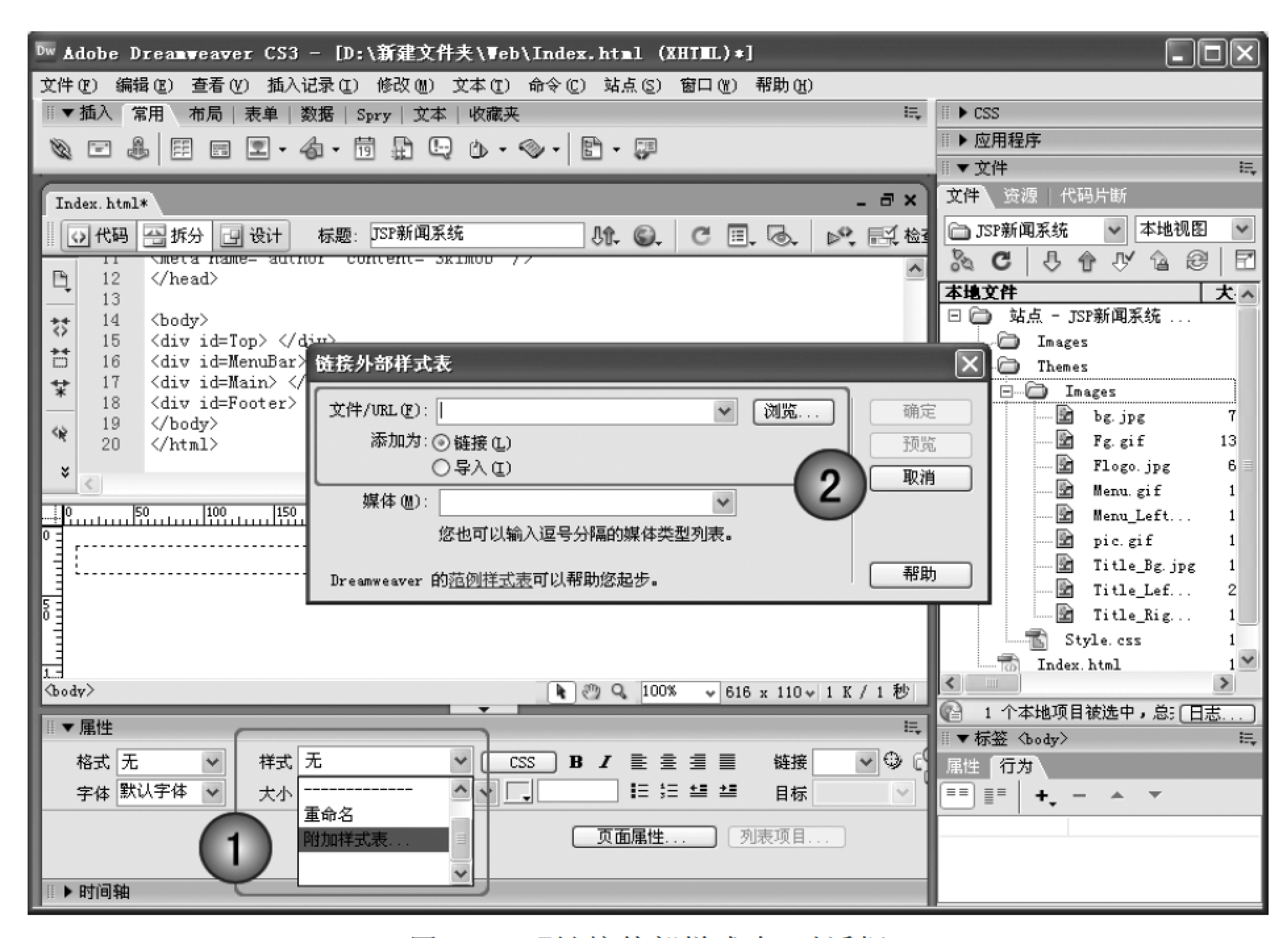


图 4-17 【链接外部样式表】对话框



钮,选择上一步建立的样式表文件 Style. css,设置完成后保存文件。

(7) 至此,完成了页面基本框架的制作,通过菜单栏的【文件】|【在浏览器中预览】命令,选择 Internet Explore,或者直接按功能键 F12 来预览 Index. html,在浏览器中的预览效果如图 4-13所示。

任务 2:页面头部的制作

对于页面头部的制作分为两个部分,左面的部分是 Logo,右面是一些附加的带有超链接的文本内容,左面 Logo 部分采用图片直接导出,设为左对齐;右面的文本部分加粗,设为右对齐,效果如图 4-18 所示。

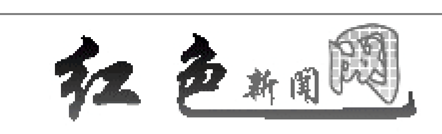


图 4-18 "新闻类别网站"页面头部的效果图

具体操作步骤如下:

(1) 在对应的位置输入如程序 4-3 所示的代码,建立 DIV 结构。

【程序 4-3】 建立 DIV 结构

Example Source Code:

```
<div id=Top>
  <div id=Logo><img src="Themes/Images/Flogo.jpg" alt="Logo" /></div>
  <div id=Banner>
    <a href="http://221.83.32.56/">学校首页</a> |
    <a onclick=defaul_home(); href="Default.jsp#" name=home>设为首页</a> |
    <a onclick=defaul_addFavorite(); href="Default.jsp#">添加收藏</a> |
    <a href="mailto: guotianbing@sxftc.edu.cn">联系我们</a>
  </div>
  <div class=clear></div>
</div>
```

(2) 在 Style. css 样式文件中添加如程序 4-4 所示的代码。

【程序 4-4】 建立相应 CSS 结构

Example Source Code:

```
#top {
    background-color: #ffffff;
    width: 880px;
    padding-right: 10px;
    padding-left: 10px;
    padding-bottom: 5px;
    padding-top: 5px;
}
#banner {
    float: right;
    margin: 0px;
    color: #990000;
    text-align: right
}
```



```
#top #logo {
    float: left
}
```

（3）预览效果。

任务 3：页脚部分的制作

对于页脚部分的制作，涉及的主要内容有文本、一些超链接，对齐方式设置为居中。页脚部分最重要的是顶部的细线，采用上边框就可实现，字体继承 body 的样式即可，效果如图 4-19 所示。

图 4-19 “新闻类别网站”页脚部分的效果图

具体操作步骤如下：

（1）在对应的位置建立如程序 4-5 所示的 DIV 结构。

【程序 4-5】 建立 DIV 结构

Example Source Code：

```
<div id=Footer>
  <div id=Copyright>
    <div>
        <a href="About.jsp">关于我们</a> |
        <a href="Readme.jsp">安装说明</a> |
        <a href="Profile.jsp">功能简介</a> |
        <a href="Copyright.jsp">版权说明</a> |
        <a href="Admin_Main.jsp">管理入口</a>
    </div>
    <div>Copyright &copy; 1998 - 2008 Tencent Inc. All Rights Reserved</div>
    <div>版权所有：@工作室</div>
  </div>
</div>
```

（2）在 Style.css 样式文件中添加如程序 4-6 所示的 CSS 结构。

【程序 4-6】 建立相应 CSS 结构

Example Source Code：

```
#footer {
    border-top: #ff0000 1px solid;
    width: 900px
}
#copyright {
    color: #cc0000;
    line-height: 200%;
    height: 50px;
    text-align: center
}
```



（3）预览效果。

任务 4：导航条的制作

导航条的制作，涉及左右两个部分，左面直接导入图片；右面的部分比较难，除了“首页”，其他的样式相同，设置超链接的 a：hover 使菜单达到动态变化的效果，在此对<a>采用浮动效果，并单独设置填充使<a>成为块状，从而实现如图 4-20 所示的效果。



图 4-20　“新闻类别网站”导航条的效果图

具体操作步骤如下：

（1）在对应的位置输入如程序 4-7 所示的代码，建立 DIV 结构。

【程序 4-7】　建立 DIV 结构

Example Source Code：

```
<div id=MenuBar>
  <div id=time>
    <script type=text/javascript>Printtime()</script>
  </div>
  <div id=Menu>
    <a id=Home href="Index.html">首页<br /><span class=Menu2>Home</span></a>
    <a href="List.html">建设动态<br /><span class=Menu2>Dynamic</span></a>
    <a href="List.html">公告通知<br /><span class=Menu2>Notice</span></a>
    <a href="List.html">他山之石<br /><span class=Menu2>Reference</span></a>
    <a href="List.html">上级文件<br /><span class=Menu2>Documents</span></a>
    <a href="List.html">师生论坛<br /><span class=Menu2>Discussion</span></a>
    <a href="List.html">工作简报<br /><span class=Menu2>Brief</span></a></div>
</div>
```

（2）在 Style.css 样式文件中添加如程序 4-8 所示的 CSS 结构。

【程序 4-8】　建立相应 CSS 结构

Example Source Code：

```
#menubar {
    width: 900px
}
#menubar #time {
    float: left
}
#menu {
    float: right
}
#menu .menu2 {
    font-size: 10px;
    line-height: normal
}
#menu #home {
    padding-left: 30px;
```



```
        background-image: url("images/Menu_left.jpg");
        background-repeat: no-repeat
    }
    #menu a {
        padding-right: 12px;
        padding-left: 12px;
        float: left;
        background-image: url("images/Menu.gif");
        padding-bottom: 12px;
        color: #ffffff;
        line-height: normal;
        padding-top: 12px;
        background-repeat: no-repeat;
        background-color: #cc0000;
        text-decoration: none
    }
    #menu a: hover {
        color: #ffff00;
        text-decoration: none
    }
```

(3) 预览效果。

4.3.3 相关知识拓展

1. CSS 层叠样式表的优先级问题

掌握了使用 CSS 层叠样式表的基本方法之后，还需要了解其有关优先级的问题。如果在页面中采用了多种样式表的定义方式，并且使用不同方式针对同一个 HTML 标记进行了重复定义，且定义的属性中有冲突，那么哪一种定义会生效呢？

例如，在外部样式表文件中对<h1>标记进行了颜色的定义，而在本页面的<style>样式中既导入了外部的样式文件，又对该标记进行了颜色的重新定义，还有可能在页面某处的<h1>标记中，使用 style 属性进行了不同颜色的定义。那么到底这几种颜色的设置到底哪个会优先生效呢？

答案是这样的：CSS 样式的定义遵从“就近一致”的原则，离所控标签越近的样式优先级越高，即使用 style 属性定义的样式优于使用<style>样式定义的内部样式，使用<style>样式定义的内部样式优于使用外部样式表文件定义的样式。故上例中，在页面某处的<h1>标记中，使用 style 属性进行的颜色定义会优先生效。

2. CSS 常用属性

CSS 常用属性如表 4-1 所示。

3. CSS 常用单位

CSS 常用单位如表 4-2 所示。



表 4-1　CSS 常用属性

属　　性	功　能	说　　明
{font-family:gill,Verdana}	字体	如果字体名含有空格,要将字体名加上双引号;已经在双引号里的要降级为单引号
{font-style: italic}	字体	italic 或 oblique 为斜体;normal 为正常体。
{font-weight: normal}	字粗细	值为 lighter、normal、bold、bolder 等,逐渐加粗;值也可以为 100、200、300 等
{font-size: 12px}	字大小	单位有 px、ex、em 等。
{color: red}	颜色	值可以为(1)red、yellow、blue 形式;(2)rgb(255,23,100)形式;(3)#rrggbb 形式
{background-color: red}	背景颜色	值同颜色设置,当值为 transparent 时,为透明
{background-image: none}	背景图	值为 none 时,表示无背景图;值为 url(图像文件的 url)时,指定背景图
{text-decoration: none}	特殊字体	值为 none: 无特殊;line-through: 删除线;underline: 下画线;overline: 上画线;blink: 闪烁等
{text-align: left}	文字对齐	值为 left、right、center、justify 等
{list-style-type: disc}	列表符样式	值为 disc(●)、circle(○)、square(■)、decimal(1,2,3)、lower-roman(i,ii,iii)、upper-roman、lower-alpha、none 等
{border-style: solid}	边框样式	值为 none(无)、dotted(点虚线)、dash(短线虚线)、solid(实线)、double(双线)、groove(3d 沟槽状)、ridge(3d 脊状)、inset(3d 内嵌)、outset(3d 外嵌)等
{border-color: red}	边框颜色	值为(1)red、yellow、blue 形式;(2)rgb(255,23,100)形式;(3)#rrggbb 形式

表 4-2　CSS 常用单位

单　位	名　称	说　　明
px	像素	pixel,是相对屏幕分辨率而言的
cm	厘米	1/100 米
mm	毫米	1/1000 米
in	英寸	1in=2.54cm
pc	派卡	1pc=1/6in
em	浏览器默认字母 M 或 H 的高度为 1em。原理:读取浏览器默认字号大小值,如 12pt;读取 em 值,如 1.5em;确定最终显示的大小为 12pt * 1.5=18pt	
pt	点	1pt=0.3478mm

4. CSS 用来定位文档元素的属性

CSS 用来定位文档元素的属性如表 4-3 所示。



表 4-3　CSS 用来定位文档元素的属性

CSS 属性	属性取值	说　明
position	absolute	绝对位置的元素是独立的
	relative	各元素位置关系是相对的
left 和 top	绝对值或百分比	定义元素的左上顶点坐标
width 和 height	绝对值或百分比	定义元素的宽和高
z-index	整数或 auto	按照 z-index 值的递增顺序由底层至顶层堆砌元素
visibility	inherit	继承：子元素在没有被定义样式的前提下，它将继承上级元素所定义的样式例如： The headline is important！ ，设置 h3 为红色，而子元素 em 没有定义颜色，它将继承 h3 的红色
	visible	元素可见
	hidden	隐藏元素

4.3.4　经验总结

1. 用 class_name 方式写类名

建议写类名时用 class_name 方式，也可以用 class-name 方式。有的教材建议使用 className 方式写类名，但这里的建议是尽量少用，因为不同浏览器对英文大小写是否敏感不得而知。

2. 样式的定义尽量使用 class 而不用 id

建议在定义样式名时尽量使用 class 而不用 id，原因有以下三点。

(1) id 不可以重用，而 class 可以重用。

(2) id 的优先级太高，若是此 id 对应的代码定义了一个 #page_content a {color：#f60}，当网页里要修改链接颜色，都必须加上 #page_content 才能越过这个优先级。

(3) id 专门留给 JS 用，这样才符合页面表现形式与动作行为分离的原则。id 命名时采用“驼峰式”。

3. margin 和 padding

使用 margin 和 padding 时，尽量省略最后一个设置值。如“margin：20px 10px 5px 10px;”，左右值是一样的，应该省略掉最后一个值，写成“margin：20px 10px 5px;”。如果以后要修改左右间距，改一个就可以了，还能避免漏改某些值。

4. 使用浮动功能时记得适当清除指令

浮动效果是个危险的功能，未必会产生设计者所期望的结果。如果遇到浮动元素延伸到外围容器的边框或者产生其他不正常情况，应先确定浮动效果的实现方法是否正确。



5. 避免同时对元素指定 padding/border 以及高度或宽度

Windows 版 IE 经常导致 width 与 height 的计算问题。有些方法可以解决此问题，但如果母元素需要指定高度与宽度时，最好能够在母元素之内的子元素套用 margin；或者当子元素需要指定高度与宽度时，在母元素套用 padding 以达效果。

6. 记住“TRBL”写法

border、margin 与 padding 的简写语法有特定顺序，从上方开始顺时针方向转动：top、right、bottom、left。所以，“margin：0 1px 3px 5px；”的结果是上方无边界，右边 1 像素，以次类推。记住“TRBL”中不要弄错字母次序。

7. 数值要指定单位

CSS 需要对每个 font、margin 等各种值指定单位，除非是 0 时可以省略。

4.4　主页、列表页和内容页的制作

4.4.1　任务描述

利用 Adobe Dreamweaver CS3 进行“新闻类别网站”主页、列表页和内容页的制作。

4.4.2　任务实现

本阶段任务可具体细分为主页的制作、列表页的制作和内容页的制作，通过以下三个任务来完成。

任务 1：主页的制作
任务 2：列表页的制作
任务 3：内容页的制作

任务 1：主页的制作

参见 2.4 节，主页的设计效果如图 2-10 所示。下面对其进行分析：除去页面的基本框架外，主页的其他部分都是在如图 2-11 所示的列表框架之内，所以主页制作的重点是在这个列表框架中。

下面对图 2-11 所示的列表框架进行详细的分析。列表框架包括上下两个部分，上部是一个圆角的矩形，并且设置了底纹，其中内容也包括左右两部分；下部是一个矩形，但是没有上边框，只有左、右、下三个边框。

具体操作步骤如下：

(1) 在对应的位置建立如程序 4-9 所示的 DIV 结构。

【程序 4-9】　建立 DIV 结构

Example Source Code：

```
<DIV class=List>
```



```
<DIV class=title>
<DIV class=title_left>图片新闻</DIV>
<DIV class=title_right><A href="">More&gt;&gt;</A></DIV>
<DIV class=clear></DIV>
</DIV>
<DIV class=connect><IMG src=" Pic/6.jpg "></DIV>
</DIV>
```

（2）在 Style.css 样式文件中添加如程序 4-10 所示的 CSS 结构。

【程序 4-10】 建立相应 CSS 结构

Example Source Code：

```
.clear{
    clear: both;
}
.List {
    margin-right: 5px;
    margin-bottom: 5px;
}
.title {
    background-image: url(Images/Title_Bg.jpg);
    background-repeat: repeat-x;
}
.title_left {
    height: 25px;
    float: left;
    background-image: url(Images/Title_Left.jpg);
    background-repeat: no-repeat;
    padding-left: 40px;
    padding-top: 10px;
    font-weight: bold;
    color: #CC0000;
}
.title_right {
    height: 25px;
    float: right;
    padding-right: 20px;
    background-image: url(Images/Title_Right.jpg);
    background-repeat: no-repeat;
    background-position: right;
    padding-top: 10px;
    font-weight: bold;
    color: #CC0000;
}
.connect {
    border-right-width: 1px;
    border-bottom-width: 1px;
    border-left-width: 1px;
    border-right-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-right-color: #CCCCCC;
```

```
    border-bottom-color: #CCCCCC;
    border-left-color: #CCCCCC;
    text-align: center;
    padding: 5px;
}
```

（3）预览效果。

任务 2：列表页的制作

参见 2.5 节，列表页的设计效果如图 2-18 所示。下面对其进行分析：除去页面的基本框架外，列表页的其他部分都是在如图 2-11 所示的列表框架之内，所以列表页制作的重点是如图 4-21 所示的新闻列表模块。

建设动态	当前位置：首页>>建设动态
[建设动态] 连铸连轧厂7月中上旬仅漏钢三次	2007-12-17
[建设动态] 生产完成进度表7-22	2007-12-14
[建设动态] 全力铸造决胜市场之剑	2007-11-22
[建设动态] 把握市场占领市场努力实现最佳利润	2007-12-24
[建设动态] 内部人才招聘启事??技术中心	2007-11-14
[建设动态] 三炼钢7#转炉炉役检修进入砌炉阶段	2007-11-20
[建设动态] 集团公司十届三次职工代表大会隆重召开	2007-11-05
[建设动态] 生产进度表7-20	2007-11-16
[建设动态] 公司企业知识仓库 使用说明	2007-11-13
[建设动态] 7月22日化验报告单	2007-12-27
[建设动态] 公司召开职代会预备会	2007-09-13
[建设动态] 生产日报7-16	2007-10-24
[建设动态] 自用坯库存表7-16	2007-10-13
[建设动态] 2004年第二季度产品质量工作总结	2007-10-20
[建设动态] 透视今年的防汛工作	2007-09-06
第 1 页　首页　前页　后页　尾页　共 19 页	

图 4-21　“新闻类别网站”列表页新闻列表模块效果图

新闻列表模块采用的是 4.3 节中制作的列表框架，重复的内容不再介绍。本节的重点是介绍底部分页的实现。

具体操作步骤如下：

（1）在对应的位置建立如程序 4-11 所示的 DIV 结构。

【程序 4-11】　建立 DIV 结构

Example Source Code：

```
<DIV class=List>
<DIV class=title>
  <DIV class=title_left><A href="http://localhost: 8080/New/">建设动态</A></DIV>
  <DIV class=title_right>当前位置: <A href="#">首页</A>><A href="#">建设动态</A>
  </DIV>
  <DIV class=clear></DIV>
</DIV>
<DIV class=connect>
  <DIV>
    <DIV class=connect_left>
```



```
        <A title=馆藏检索 href="http://localhost: 8080/New/List.jsp?MenuID=25">[建设动态]
        </A>
        <A title=第二季度产品质量 href=" Content.jsp?ID=1592">第二季度产品质量工作</A>
      </DIV>
      <DIV class=connect_right>2007-10-20</DIV>
      <DIV class=clear></DIV>
    </DIV>
第 1 页  <FONT color=gray>首页  前页</FONT>  <A
href="List.jsp?MenuID=25&amp;Page=2">后页</A>  <A
href="List.jsp?MenuID=25&amp;Page=19">尾页</A>  共 19 页
</DIV>
</DIV>
```

(2) 在 Style.css 样式文件中添加如程序 4-12 所示的 CSS 结构。

【程序 4-12】 建立相应 CSS 结构

Example Source Code:

```
.connect_right {
    float: right;
    color: #990000;
}
.connect_left {
    float: left;
    list-style-position: outside;
    list-style-image: url(Images/pic.gif);
    list-style-type: square;
}
```

(3) 预览效果。

任务 3：内容页的制作

参见 2.5 节，内容页的设计效果如图 4-19 所示。下面对其进行分析：除去页面的基本框架外，内容页的其他部分都是在如图 2-11 所示的列表框架之内，所以内容页制作的重点是图 2-20 所示的内容显示模块。

图 2-20 中，内容页的内容显示模块可进一步划分，具体分为五块：标题、相关信息、导读、内容和前后导航，在套用了实际数据的显示中，这种结构依然不变，只是显示的具体内容发生了变化。对这五个部分分别进行显示样式的控制，定义五种样式并详细设置相应的属性，以达到预期的设计效果。

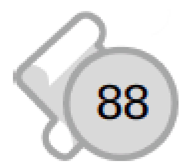
具体操作步骤如下：

(1) 参见图 4-22，在对应的位置建立如程序 4-13 所示的 DIV 结构。

【程序 4-13】 建立 DIV 结构

Example Source Code:

```
<DIV class=List>
<DIV class=title>
<DIV class=title_left>公告通知</DIV>
<DIV class=title_right>当前位置：首页 &gt;&gt;公告通知</DIV>
<DIV class=clear></DIV></DIV>
```



公告通知　　　　当前位置：首页>>公告通知

退休职工马振山举办个人庭院画展

发布时间：2007-03-11 被阅览数：1502 次 作者：按时地方 来源：地方

导读：时殷弘说，谈奥巴马到现在为止已经表示的对外政策主张，我觉得他同共和党的政治纲领间差别非常大，麦凯恩虽然最终要撤，但是是漫长的过程。但是奥巴马多次许诺，他要迅速撤军。

在谈到朝鲜核问题时，他说，相信奥巴马一定会尝试用美朝柏林会谈以来，比布什政府更大的外交努力，就是通过对朝鲜做让步来争取半岛局势的改善。

时殷弘最后说，美国的权势、力量、影响肯定在衰弱，当然这个衰弱到底有多大？至少最保守说，已经发生了。但是奥巴马上台肯定是提供了一个可能性，大家都说奥巴马是一个在对外政治上是一个世界大部分国家都基本满意且比较喜欢的总统，我想不一定。

上一篇文章：我欲称道总关情（创刊30周年征…　　下一篇文章：检测3721的小工具

图 4-22　“新闻类别网站”内容页内容显示模块效果图

```
<DIV class=connect>
<DIV id=Title>退休职工马振山举办个人庭院画展</DIV>
<DIV class=Xin><STRONG>发布时间：</STRONG>2007-03-11 <STRONG>被阅览数：</STRONG>1502 次
<STRONG>作者：</STRONG>按时地方 <STRONG>来源：</STRONG>地方</DIV>
<DIV class=Dao>
<p><strong>导读：</strong>时殷弘说，谈奥巴马到现在为止已经表示的对外政策主张，我觉得他
同共和党的政治纲领间差别非常大，麦凯恩虽然最终要撤，但会是漫长的过程。奥巴马多次许诺，他
要迅速撤军。</p></DIV>
<DIV class=Read>
<P>在谈到朝鲜核问题时，他说，相信奥巴马一定会尝试用美朝柏林会谈以来，比布什政府更大的外
交努力，就是通过对朝鲜做让步来争取半岛局势的改善。</P>
<p>时殷弘最后说，美国的权势、力量、影响肯定在衰弱，当然这个衰弱到底有多大？至少最保守说，
已经发生了。但是奥巴马上台肯定是提供了一个可能性，大家都说奥巴马是一个在对外政治上是一
个世界大部分国家都基本满意且比较喜欢的总统，我想不一定。</p></DIV>
<DIV>
<DIV class=Dao>
<DIV class=connect_left>上一篇文章：<A title=我欲称道总关情(创刊 30 周年征文)
href="http://localhost:8080/New/Content.jsp?ID=1402">
我欲称道总关情(创刊 30 周年征…</A></DIV>
<DIV class=connect_right>下一篇文章：<A title=检测 3721 的小工具
href="http://localhost:8080/New/Content.jsp?ID=1404">检测 3721 的小工具</A></DIV>
<DIV class=clear></DIV>
</DIV>
</DIV>
</DIV>
</DIV>
```

（2）在 Style.css 样式文件中添加如程序 4-14 所示的 CSS 结构。

【程序 4-14】　建立相应 CSS 结构

Example Source Code：

```
#Title {
    font-size: 16px;
    font-weight: bold;
```



```
        color: #990000;
        text-align: center;
        margin-bottom: 10px;
    }
    .Xin {
        font-size: 12px;
        text-align: center;
        line-height: normal;
    }
    .Dao {
        border: 1px dashed #cccccc;
        margin: 20px;
        background-color: #efefef;
        text-align: left;
    }
    .Read {
        text-align: left;
    }
    .clear{
        clear: both;
    }
    .connect_right {
        float: right;
        color: #990000;
    }
    .connect_left {
        float: left;
        list-style-position: outside;
        list-style-image: url(Images/pic.gif);
        list-style-type: square;
    }
```

（3）预览效果。

4.4.3 相关知识拓展

1. 扩展的 CSS 支持

在 Dreamweaver MX 2004 中，CSS 功能已经非常强大。而 Dreamweaver CS3 在 CSS 方面支持更多。Dreamweaver CS3 增强了复杂样式表信息的显示，减少了必须跳转到网页浏览器来检查代码设计的次数。

Dreamweaver CS3 有几个可视化设置提供了非常有用的查看方式，从而解决了查看复杂 CSS 布局这一难题。CSS 面板现被设计为一个统一的面板，将 Dreamweaver MX 2004 中的众多 CSS 面板集中到了一个位置，这样 CSS 面板就变成了一个更富有可用性的控制面板。使用 CSS 面板可以快速确认样式、编辑样式、查看应用于页面元素的样式（就像查看段落、图像和链接一样）。

Dreamweaver CS3 在 CSS 面板中加入了一个摘要视图，用于快速确认应用于当前选定





元素的 CSS 属性。

执行【窗口】|【CSS 样式】命令，可以打开【CSS 样式】面板，单击【添加属性】图标，出现一个空白文本框，在该框的右边单击向下箭头，在弹出的下拉列表中可以选择要设置的新样式属性，比如选中 color，就会在属性列中添加一个 color 属性行，然后在该属性行后面的颜色选择器中选择一种颜色就可以了，如图 4-23 所示。这比起以前要反复修改网页代码要方便很多。



图 4-23　添加 CSS 属性

单击【CSS 样式】面板中的【正在】按钮，可以显示当前所选中内容的 CSS 摘要，而与所选内容无关的 CSS 会以删除线的形式出现。这个功能对快速确认应用于当前选定元素的 CSS 属性很有帮助，在列表中可以方便地查看哪种样式中的哪种属性被应用到当前选中的元素中，如图 4-24 所示。



图 4-24　【CSS 样式】面板



2. 其他新功能

(1) Dreamweaver CS3 中增加了一个代码折叠的新功能，这个功能为编码人员提供了方便。HTML 代码有时长得可以翻好几页，如果程序员在编写一段代码时想查看另一段代码，就得用滚动条翻来翻去，终于找到了这段代码，然后又得用滚动条翻回去，除非在开始查找时，事先定义一个特殊的标记以便返回原处。Dreamweaver CS3 现在解决了这一问题，相关的代码段可以通过单击代码窗口左边【代码】工具栏的图标按钮来展开和关闭。例如在【代码】视图中选中其中一个<table>与</table>之间的编码，如图 4-25 所示，然后单击显示的减号，就可将选中的代码折叠，如图 4-26 所示。

E:\WZ\HZ\MOBAN.html (XHTML)

```
    <td><img src="images/HZ_7.gif" width="84" height="75" /></td>
    <td><img src="images/HZ_8.gif" width="85" height="75" /></td>
    <td><img src="images/HZ_9.gif" width="71" height="75" /></td>
    <td><img src="images/HZ_10.gif" width="33" height="75" /></td>
  </tr>
</table>
<table width="722" height="178" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td height="178" valign="top" background="images/HZ_11.gif"> </td>
  </tr>
</table>
<table width="722" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td><img src="images/HZ_12.gif" width="233" height="285" /></td>
    <td width="456" valign="top"> </td>
    <td><img src="images/HZ_14.gif" width="33" height="285" /></td>
```

<body> <table> 1 K / 1 秒

图 4-25 折叠代码

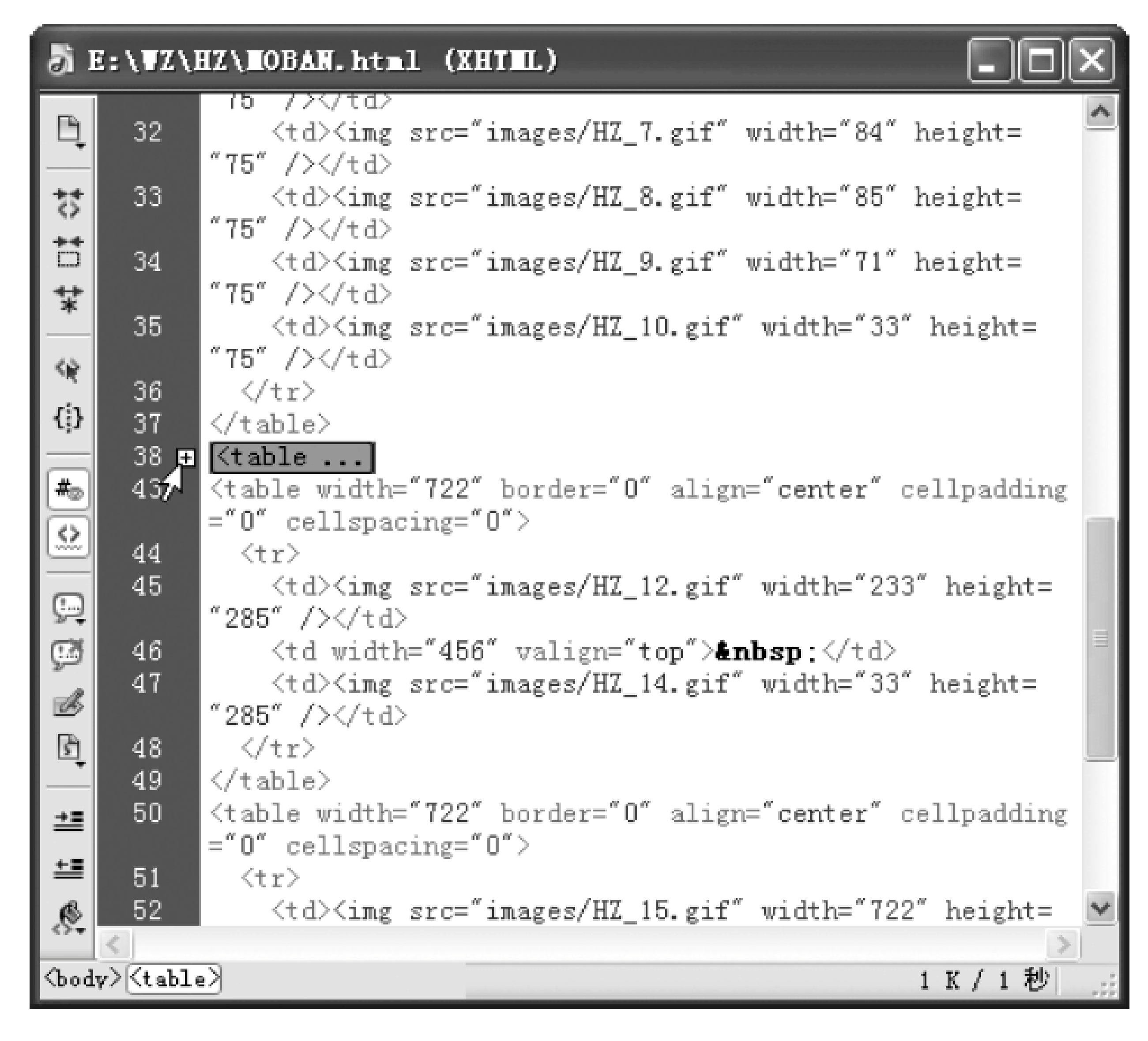


图 4-26 展开代码



（2）Dreamweaver CS3 的其他新功能还包括文件传输功能、支持 XML/XSLT、工作空间布局和文件比较等。

- 文件传输功能：在使用 Dreamweaver CS3 以前的版本向服务器传送文件时，不能在 Dreamweaver 中进行其他的工作。新版的 Dreamweaver 允许在传输文件的同时进行其他工作，这样可以大大提高工作效率。
- 支持 XML/XSLT：新的 XML/XSLT 创建功能简化了用于 Web 浏览的 XML 文件的格式化过程。可以创建 XSLT 文件，并完全使用 CSS 格式，将其转换成难以理解的 XML 文件，附加到本地文件中，也可以附加到互联网上的远端文件中。
- 工作空间布局：可以定制和保存工作空间配置。Dreamweaver CS3 附带 4 种不同的配置以满足设计者和编码人员的需求。可以保存和恢复不同的面板设置，以便为不同的活动自定义工作区。当保存工作区布局时，Dreamweaver 会记住指定布局中的面板以及其他属性，例如面板的位置和大小、面板的展开或折叠状态以及应用程序窗口的位置和大小等。
- 文件比较：Dreamweaver CS3 可以使用文件比较工具（也称为“diff 工具”），比较同一文件的本地和远程版本的代码、两个不同的远端文件的代码、两个不同的本地文件的代码。如果在本地处理过某个文件并怀疑该文件在服务器上的副本已被他人篡改时，比较本地和远程版本的操作就十分有用。可以在将文件上传到服务器之前查看远程更改并将这些更改合并到本地版本中，而无须离开 Dreamweaver。
- 当然，Dreamweaver 还有另外其他的新增功能和改进，在后面的内容中，会结合要讲述的相关内容逐一向读者介绍。

4.4.4　经验总结

在传统的表格布局中，可以对表格应用对齐方式实现网页布局。而应用 Web 标准构建网页后，float（浮动）属性也成了布局中非常重要的属性，通过对 DIV 元素应用 float 属性进行布局，不但可以实现对整个版式的规划，还可以对一些基本元素如导航等进行排列。

float 属性的值指出了对象是否浮动及如何浮动。当该属性不等于 none 值时，引起对象浮动，此时对象将被视做块对象（block-level），即该对象的 display 属性等于 block。也就是说，浮动对象的 display 特性将被忽略。

float 属性的取值有：none 表示对象不浮动；left 表示对象浮在左边；right 表示对象浮在右边。

4.5　实　　训

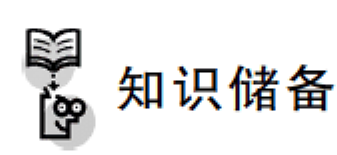
利用 Adobe Dreamweaver CS3 完成“新闻类别网站”框架页面的制作，并制作网站首页、列表页和内容页等主要页面。



第5章　构建网站的开发环境

“工欲善其事，必先利其器”，要进行项目的开发，选择一个合适的开发环境及工具是一个项目成功的前提。“新闻类别网站”采用 Adobe 公司的 Dreamweaver CS3 作为前台页面的制作工具，采用免费的 Tomcat 作为应用服务器，同样是免费而且开源的 Eclipse 作为开发工具，数据库选用微软公司的 SQL Server 2005。本章着重讲解这些软件的安装、配置和整个环境的协调配置。

5.1　安装 JDK



知识储备

JSP 技术目前唯一支持的脚本语言为 Java 语言，在 JSP 页面运行时，需要有相应的编译器编译和解释器来执行这些 Java 代码。执行 Java 代码，需要下载和安装 Java 的 JDK 开发工具包。JDK（Java Development Kit，Java 开发包，Java 开发工具）是一个编写 Java Applet 和 Java 应用程序的软件开发环境，它由一个处于操作系统层之上的运行环境以及开发者编译、调试和运行用 Java 语言编写的 Applet 和应用程序所需的工具组成。在计算机上，安装了 JDK 开发工具包就可以编译和运行 Java 代码了。JDK 是 Sun Microsystems 公司针对 Java 开发人员而开发的产品。自从 Java 推出以来，JDK 已经成为使用最广泛的 Java SDK（Software Development Kit）。

5.1.1　任务描述

构建“新闻类别网站”的开发环境任务之一：安装 Java Development Kit（简称 JDK）。要运行 Java 程序，就必须安装 JDK，JDK 是整个 Java 的核心，其中包括 Java 编译器、JVM（Java Virtual Machine）、大量的 Java 工具以及 Java 基础 API。读者可以从 http://Java.sun.com 下载 JDK。

5.1.2　任务实现

具体操作步骤如下：

（1）登录 Sun 公司的官方站点 http://java.sun.com，下载 JDK 1.6.0_05 Windows

版：jdk-6u5-windows-i586-p. exe，将其保存在本地机合适的路径下。

（2）双击 jdk-6u5-windows-i586-p. exe，启动 JDK 安装向导，如图 5-1 所示，稍等一会儿将进入下一界面。



图 5-1　JDK 1. 6. 0_05 安装向导

（3）出现如图 5-2 所示的安装许可证协议，单击【接受】按钮。

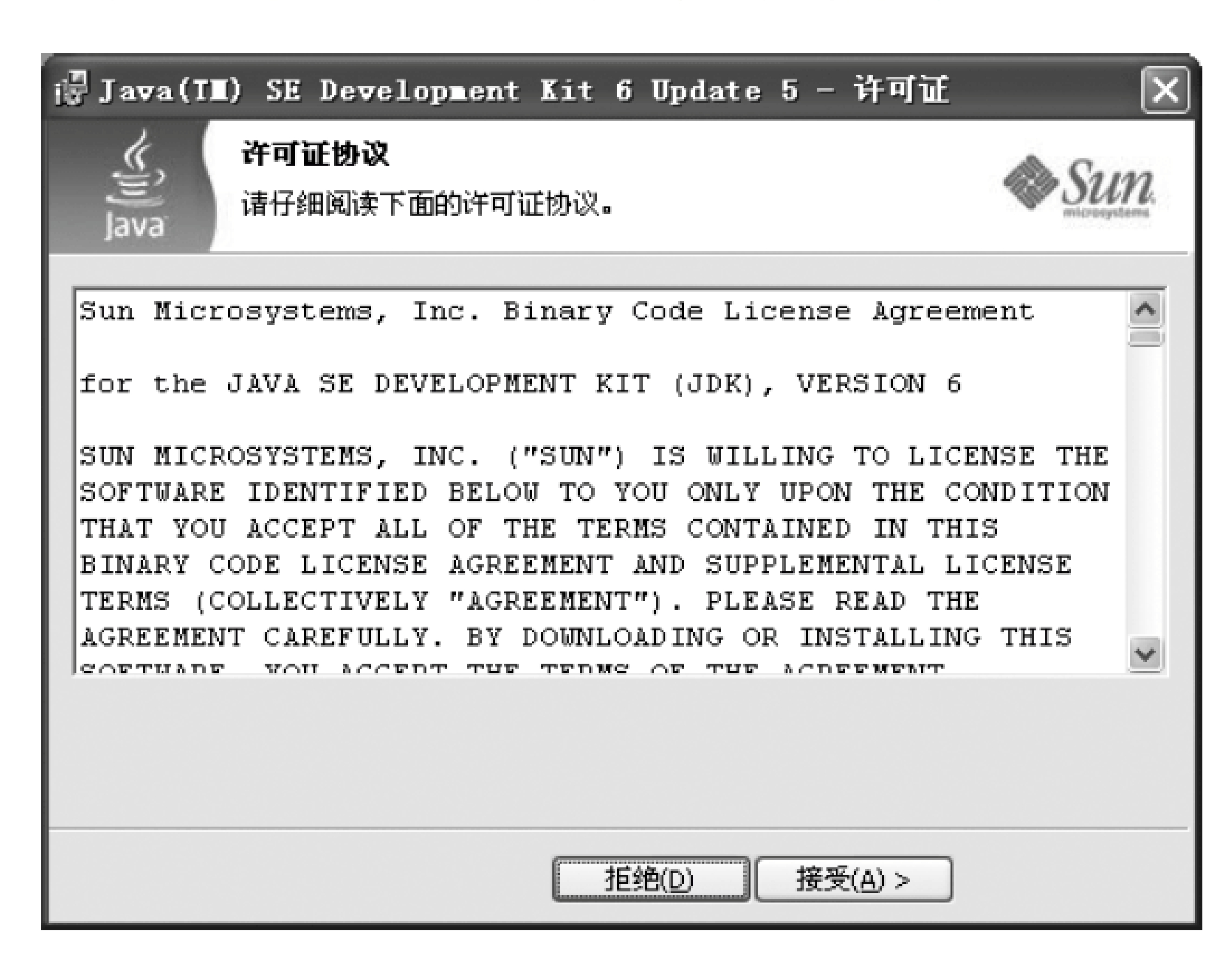


图 5-2　安装协议

（4）在【自定义安装】界面选择要安装的功能，如图 5-3 所示。可按默认设置将 JDK 安装至 C：\Program Files\Java\jdk1. 6. 0_05 路径下，也可单击【更改】按钮重设安装目录。设置好后单击【下一步】按钮，开始安装 JDK 1. 6. 0_05，屏幕将显示 JDK 的安装进度界面，安装过程可能需要持续几分钟时间，请读者耐心等待。

（5）JDK 安装完成后，接着安装 JRE1. 6. 0_05，与安装 JDK 类似，可以选择要安装的 JRE 功能，修改 JRE 的安装路径，如图 5-4 所示。单击【下一步】按钮出现 JRE 安装进度，安装可能会花费几分钟，请读者耐心等待。

（6）安装后将出现如图 5-5 所示的完成界面，请读者注意自己的安装过程是否有误。





图 5-3　选择 JDK 安装方式并设定安装目录

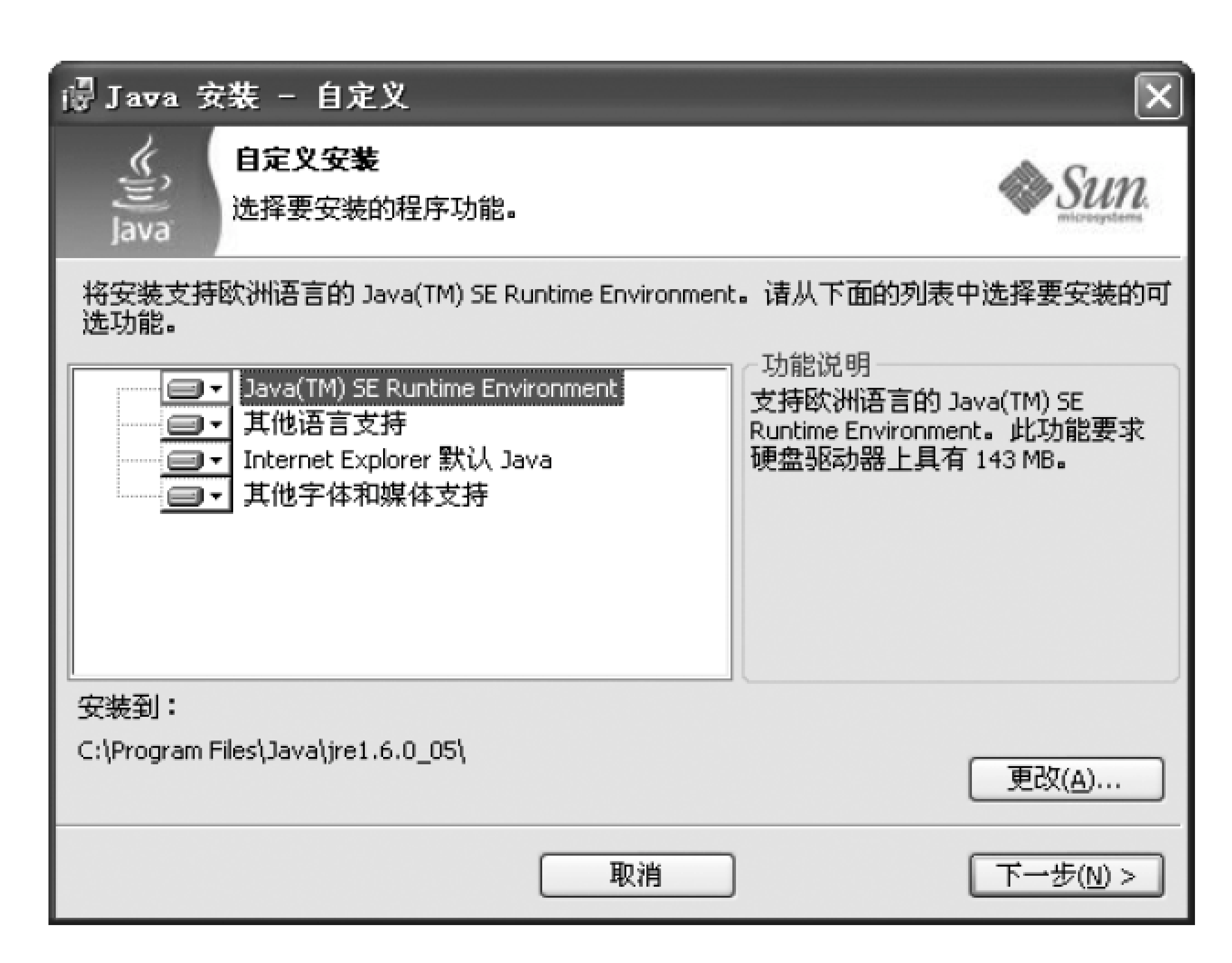


图 5-4　选择 JRE 安装方式并设定安装目录



图 5-5　JDK 安装完成



（7）选择桌面左下角的【开始】|【运行】命令，输入 cmd，调出 DOS 命令符操作界面，在光标处输入：java，显示如图 5-6 所示的内容，表示 JDK 安装成功。

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>java
Usage: java [-options] class [args...]
           (to execute a class)
   or  java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
    -client       to select the "client" VM
    -server       to select the "server" VM
    -hotspot      is a synonym for the "client" VM  [deprecated]
                  The default VM is client.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                  A ; separated list of directories, JAR archives,
                  and ZIP archives to search for class files.
    -D<name>=<value>
                  set a system property
    -verbose[:class|gc|jni]
                  enable verbose output
    -version      print product version and exit
    -version:<value>
                  require the specified version to run
    -showversion  print product version and continue
    -jre-restrict-search | -jre-no-restrict-search
                  include/exclude user private JREs in the version search
    -? -help      print this help message
```

图 5-6　JDK 测试

5.1.3　相关知识拓展

1．JVM

JVM (Java Virtual Machine)即 Java 虚拟机，它是一个虚构出来的计算机，是可运行 Java 代码的假想计算机，它通过在实际的计算机上仿真模拟各种计算机功能来实现，只要根据 JVM 规格描述将解释器移植到特定的计算机上，就能保证经过编译的任何 Java 代码能够在该系统上运行。Java 虚拟机有完善的硬件架构，如处理器、堆栈、寄存器等，还具有相应的指令系统。JVM 屏蔽了与具体操作系统平台相关的信息，使得 Java 程序只需生成到 Java 虚拟机上运行的目标代码(字节码)，就可以在多种平台上不加修改地运行。Java 虚拟机在执行字节码时，实际上还是把字节码解释成具体平台上的机器指令执行。JVM 体系结构如图 5-7 所示。

2．JRE

JRE(Java Runtime Environment)即 Java 运行环境，运行 Java 程序所必需的环境集合，包含 JVM 标准实现及 Java 核心类库。

JRE 包括 Java Runtime Environment 和 Java Plug-in JavaRuntime Environment(JRE)，是运行、测试和传输应用程序的 Java 平台。它包括 Java 虚拟机、Java 平台核心类和支持文件，不包含开发工具(编译器、调试器和其他工具)。JRE 需要辅助软件——JavaPlug-in，以便在浏览器中运行 Applet。



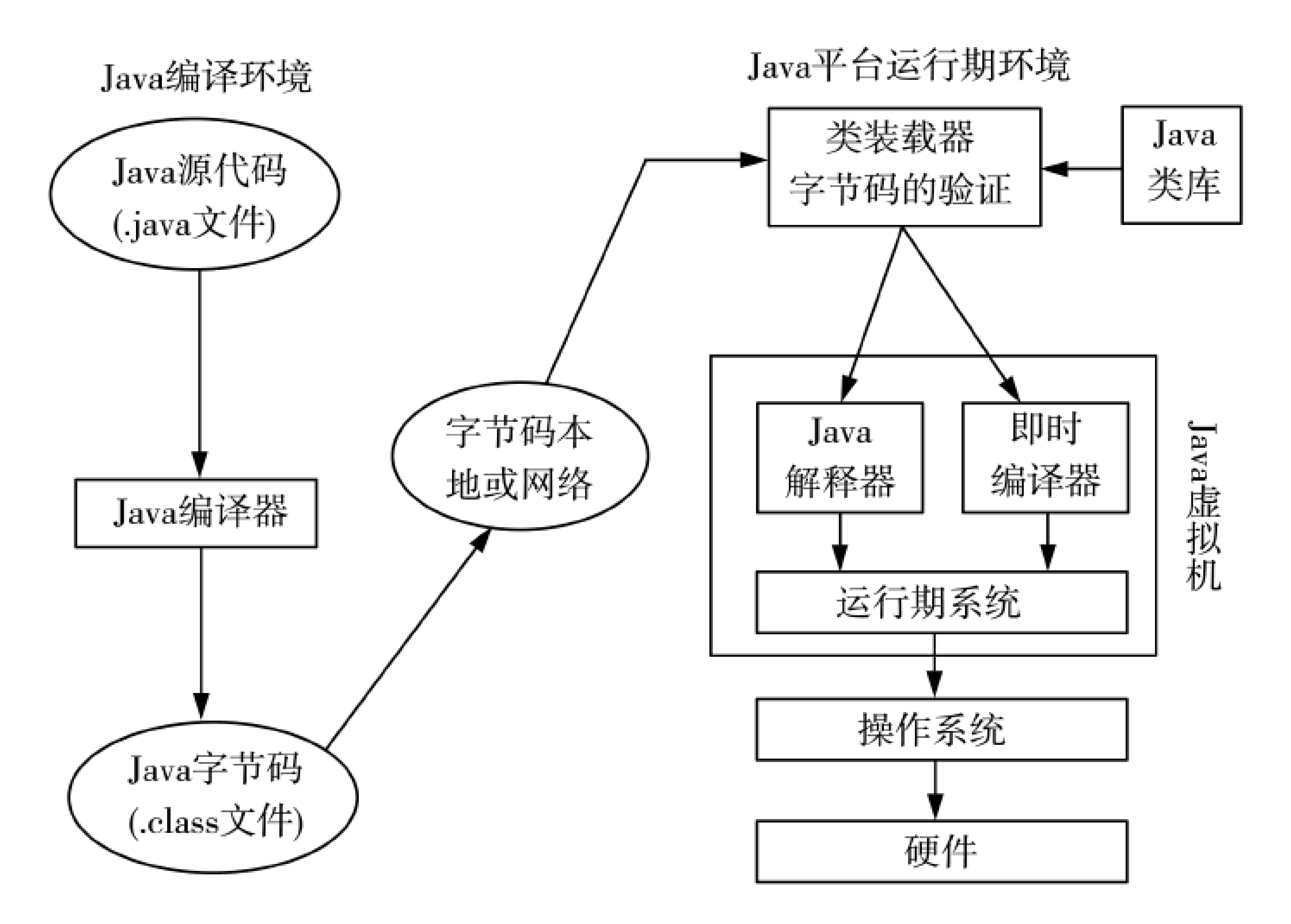


图 5-7　JVM 体系结构

JavaPlug-in 软件允许 JavaApplet 和 JavaBeans 组件在使用 Sun 的 Java Runtime Environment(JRE)的浏览器中运行,而不是在使用默认的 Java 运行环境的浏览器中运行。JavaPlug-in 可用于 NetscapeNavigator 和 Microsoft Internet Explorer。

3. JDK

JDK(Java Development Kit)是 Sun Microsystems 针对 Java 开发的产品。自从 Java 推出以来,JDK 已经成为使用最广泛的 Java SDK(Software Development Kit)。

JDK 是整个 Java 的核心,包括了 Java 运行环境(Java Runtime Environment)、Java 工具和 Java 基础的类库(rt. jar)。不论什么 Java 应用服务器,实质都是内置了某个版本的 JDK,因此掌握 JDK 是学好 Java 的第一步。最主流的 JDK 是 Sun 公司发布的 JDK,除了 Sun 之外,还有很多公司和组织都开发了自己的 JDK,例如 IBM 公司开发的 JDK,BEA 公司的 Jrocket,还有 GNU 组织开发的 JDK 等。其中 IBM 的 JDK 包含的 JVM(Java Virtual Machine)运行效率要比 Sun JDK 包含的 JVM 高出许多。而专门运行在 X86 平台的 Jrocket 在服务端运行效率也要比 Sun JDK 好很多。

从 SUN 的 JDK5.0 开始,提供了泛型等非常实用的功能,其版本信息也不再延续以前的 1.2、1.3、1.4,而是变成了 5.0、6.0 了。并且从 6.0 开始,其运行效率得到了非常大的提高,尤其是在桌面应用方面。

JDK 是 Java 开发工具包,基本上每个学 Java 的人都会先在机器上装一个 JDK,JDK 的安装目录下有 6 个文件夹、一个 src 类库源码压缩包、其他几个声明文件。其中,真正在运行 Java 时起作用的是以下 4 个文件夹:bin、include、lib、jre。可以看出这样一个关系,JDK 包含 JRE,而 JRE 包含 JVM。

bin 文件夹下最主要的是编译器(javac. exe),include 文件夹下是 Java 和 JVM 交互用的头文件,lib 文件夹下是 Java 类库,jre 文件夹下是 Java 运行环境。

注意

这里的 bin、lib 文件夹和 jre 里的 bin、lib 是不同的。总的来说，JDK 是用于 Java 程序的开发，而 jre 文件夹下的内容只能运行 class 文件，而没有编译的功能。

要运行 Java 程序，必须安装 JDK，JDK 是整个 Java 的核心，其中包括 Java 编译器、JVM、大量的 Java 工具以及 Java 的基础 API，可以从 http://Java. sun. com 中下载 JDK。JDK 包含的几个重要的命令是：

- Java 执行工具，是启动 JVM(Java 虚拟机)并执行 class(BYTE CODE，字节码)文件的命令。
- Javac 编译器，由.java 文件生成.class 文件。
- Jar Java 压缩打包工具。
- Javadoc 文档生成器。

5.1.4　经验总结

本节讲述的是采用安装程序直接安装 JDK，不需要进行设置。而实际安装中还有一种情况，就是直接将程序拷入或解压到指定位置，然后进行相应的设置。

j2sdk 可以安装在 C:\根目录下，当然 JDK 的安装路径可以选择任意磁盘目录，不过建议存放用的目录层次设置得浅一点，如果目录层次很深，下面的环境变量的配置就要麻烦很多。JDK 环境变量的具体设置如下。

1. 添加 JAVA_HOME 环境变量

右击【我的电脑】，从弹出的菜单中选择【属性】|【高级】|【环境变量】|【新建】选项，在变量名中输入 JAVA_HOME，在变量值中输入值(此处假定 JDK 安装在 C：\jdk 目录下)：

```
C: \ jdk
```

该变量值就是 JDK 的安装目录，一些 Java 版的软件和 Java 的工具需要用到该变量，设置 path 和 classpath 的时候，也可以使用该变量以方便设置。

2. 添加 path 环境变量

按同样的方法新建环境变量 path，在变量值中输入值：

```
%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;
```

其中，%JAVA_HOME%是 JDK 的安装路径。path 变量用于指定一个路径列表，搜索可执行文件。执行一个可执行文件时，如果该文件不能在当前路径下找到，则依次寻找 path 中的每一个路径，直至找到，如果找完 path 中的路径也不能找到则报错。Java 的编译命令(javac)、执行命令(java)和一些工具命令(javadoc，jdb 等)都在其安装路径下的 bin 目录中，因此应该将该路径添加到 path 变量中。



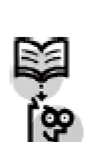
3. classpath 环境变量添加

按同样方法新建环境变量 classpath，在变量值中输入值：

```
.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;
```

特别要注意最前面的是“.;”，这个是告诉 JDK，搜索 class 时先查找当前目录的 class 文件，classpath 环境变量是引入类库，指定一个路径列表，用于搜索 Java 编译或者运行时需要用到的类。在 classpath 列表中除了可以包含路径外，还可以包含.jar 文件。Java 查找类时会把这个.jar 文件当做一个目录来进行查找。有时也需要把 JDK 安装路径下的 jre\lib\rt.jar(Linux：jre/lib/rt.jar)包含在 classpath 中。

5.2 安装、配置 Tomcat



知识储备

自从 JSP 发布以后，就出现了各式各样的 JSP 服务器，本教材选择了 Tomcat，并且由于 Windows 操作系统被广泛使用及其用户界面较友好、操作方便的特点，故选择在 Windows 版本下运行的 Tomcat 版本。Tomcat 是一种简单的 JSP 服务器，可从 http://tomcat.apache.org/ 中免费下载。

Tomcat 也是一个免费的开源 Servlet 容器，它是 Apache 基金会的 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总能在 Tomcat 中得到体现。Tomcat 是完全由 Java 编写的，其 Win32 版本与 Linux 版本的安装没有多大的区别。所以虽然下面以 Tomcat 的 Win32 版本在 Windows XP 中的安装为例说明其安装过程，但对于 Tomcat 的 Linux 版本的安装亦具有指导意义。

5.2.1 任务描述

构建“新闻类别网站”的开发环境之二：安装 Tomcat。

5.2.2 任务实现

具体操作步骤如下：

(1) 下载 Tomcat 的安装程序 apache-tomcat-6.0.18.exe 后，双击启动该安装程序，出现如图 5-8 所示的 Tomcat 安装向导界面，在该界面上单击【Next】按钮。

> **注意**
> 在安装 Tomcat 之前，首先要安装 JDK。





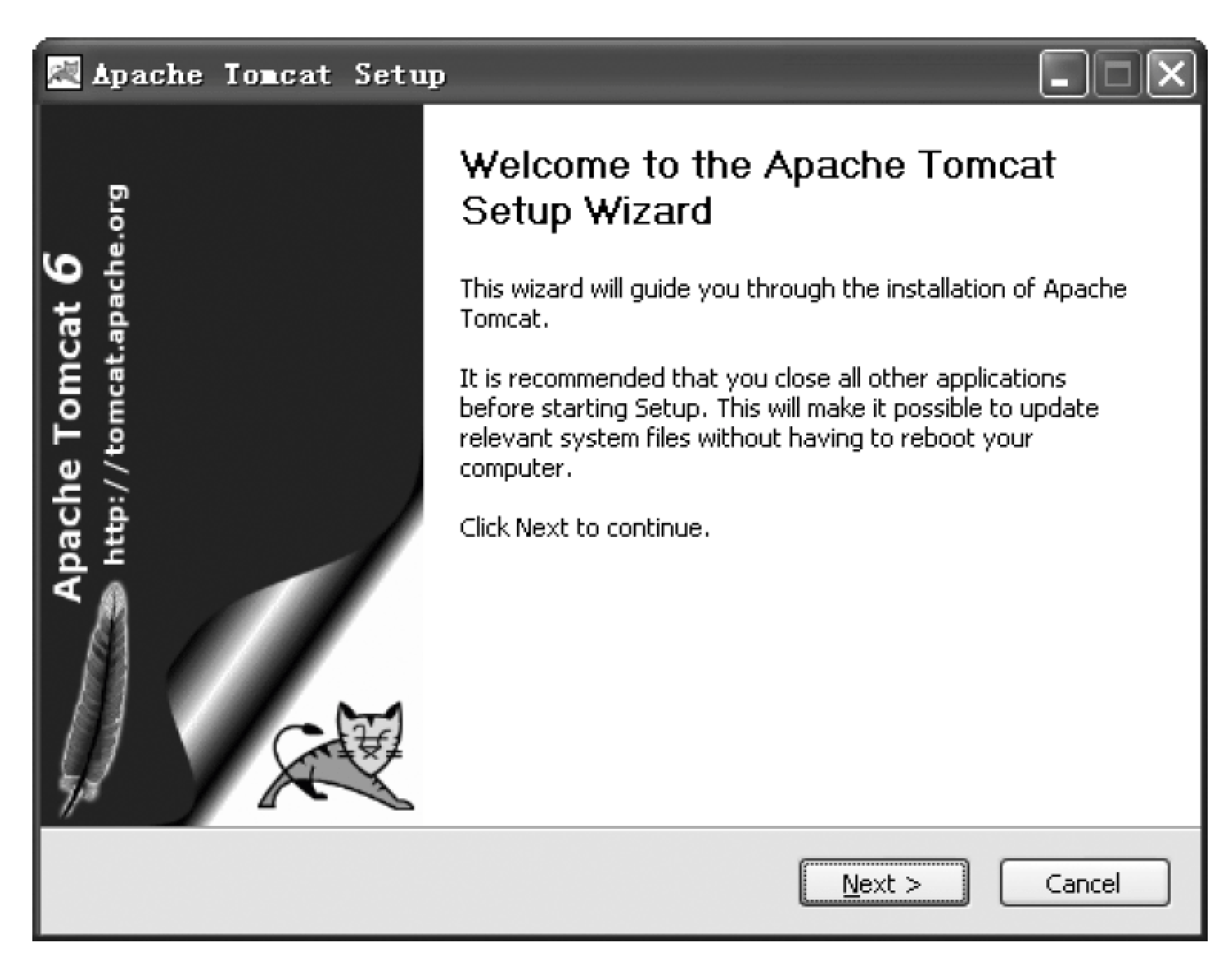


图 5-8　进入 Tomcat 安装向导

（2）出现如图 5-9 所示的 Tomcat 安装协议界面，单击【I Agree】按钮，表示同意安装协议。

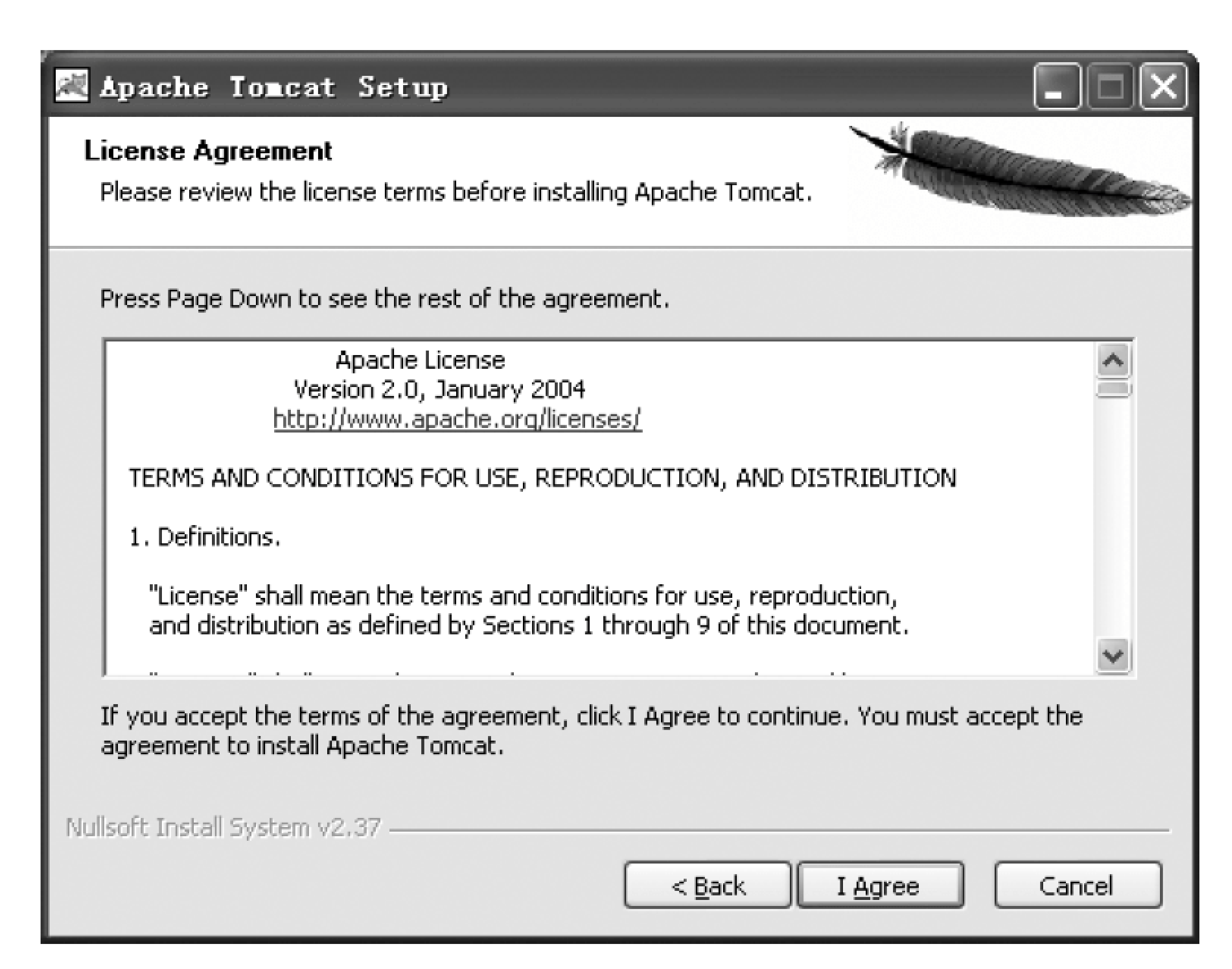


图 5-9　安装协议

（3）出现如图 5-10 所示的选择安装方式界面，选择要安装的功能，也可以采用 Tomcat 的默认安装，然后单击【Next】按钮。

（4）出现如图 5-11 所示的设置安装路径界面，可以采用默认的安装路径，也可以单击【Browse】按钮修改为其他路径，然后单击【Next】按钮。





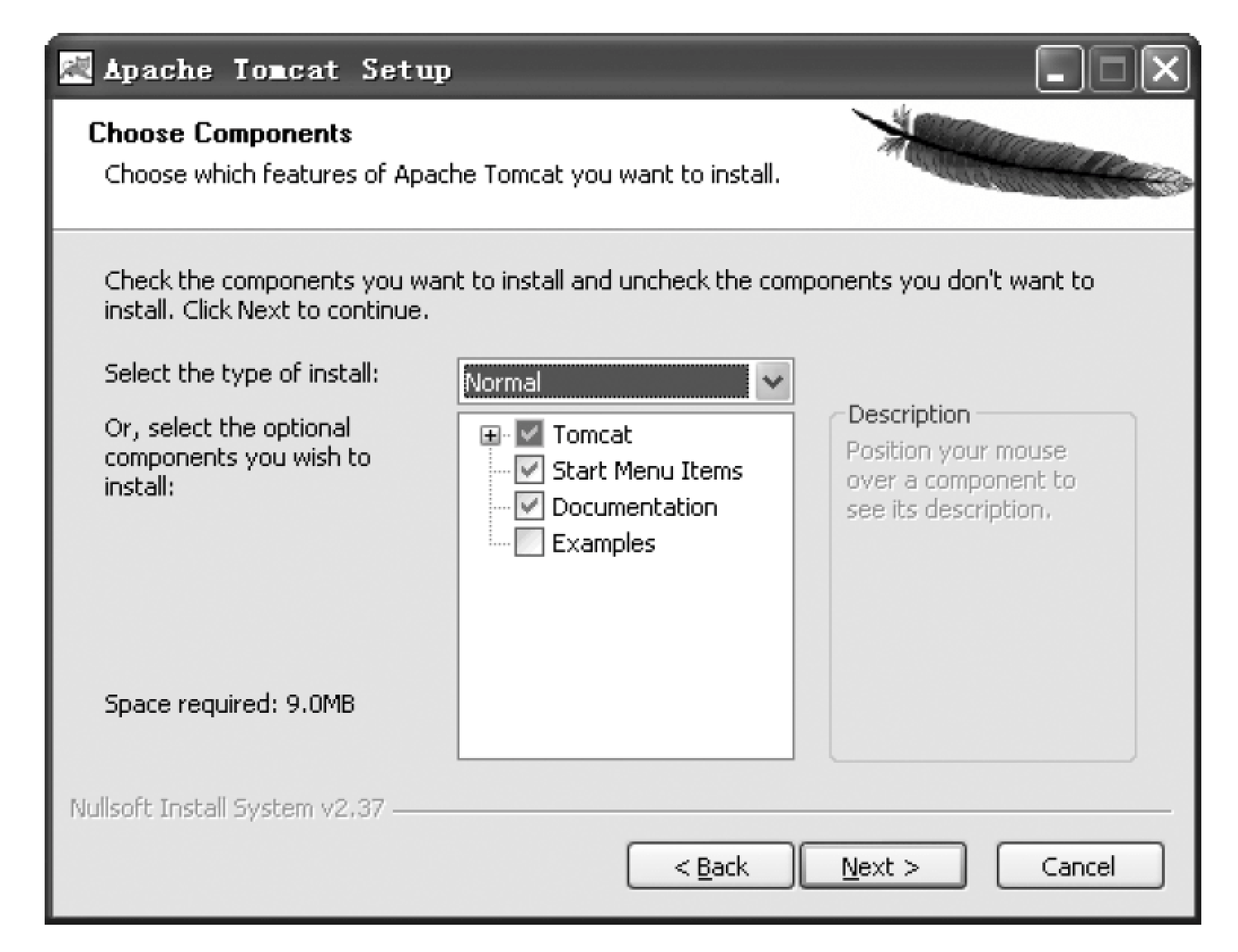


图 5-10　选择安装方式

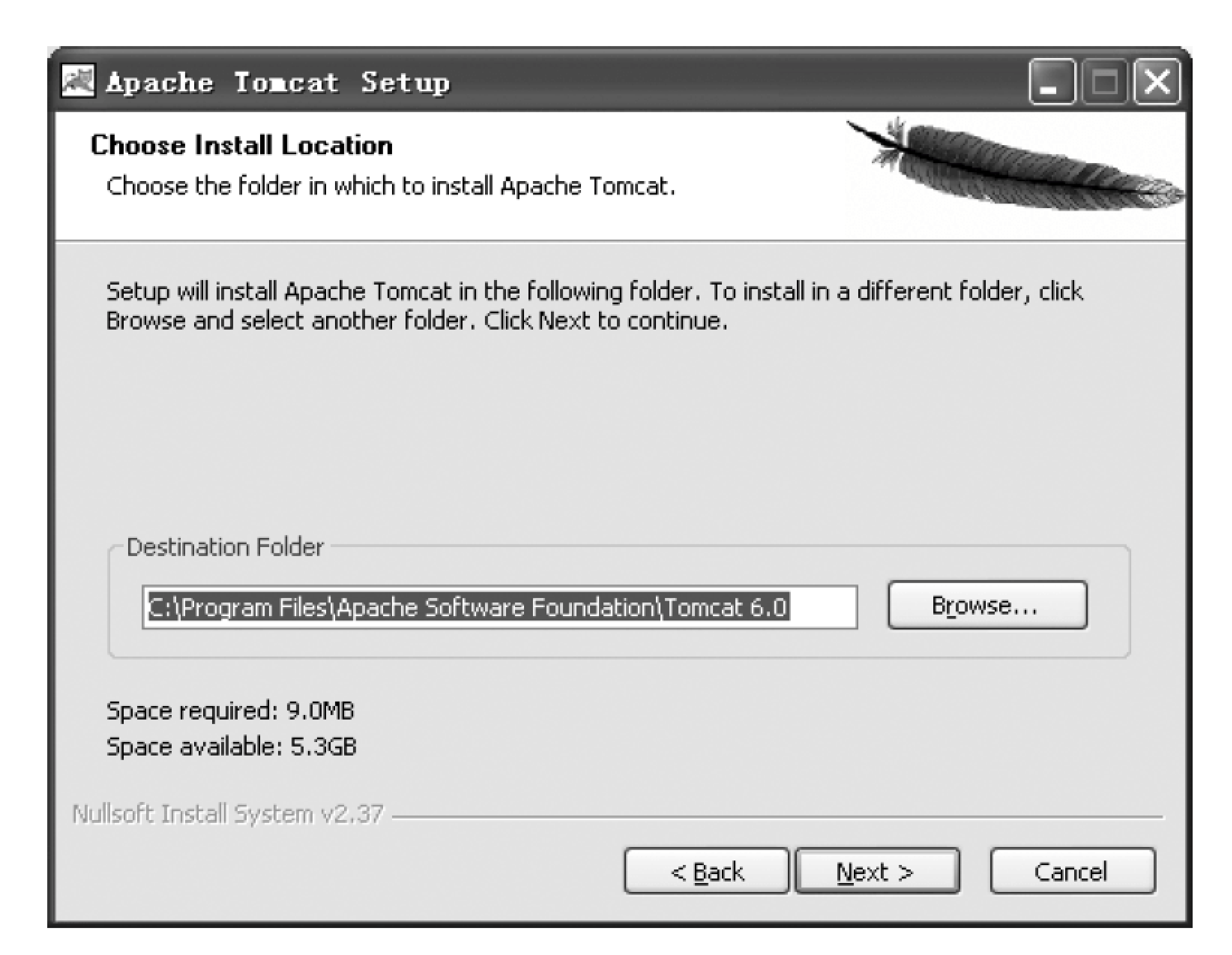


图 5-11　选择安装路径

（5）出现如图 5-12 所示的界面，此界面用于对 Tomcat 进行基本配置，包括设置 Tomcat 使用的 HTTP 端口号以及 Web 管理界面的用户名和密码。注意 HTTP 连接端口号，该端口号是客户连接到 Tomcat 时所使用的端口，应确保该端口未被其他程序占用。设置完成后单击【Next】按钮。

（6）出现如图 5-13 所示的界面，此界面用于选择 Java 虚拟机。安装程序会自动搜索安装 JDK 时安装的 Java 虚拟路径，如果没有正确显示，可以手工修改。设置完成后单击【Install】按钮，开始安装 Tomcat。



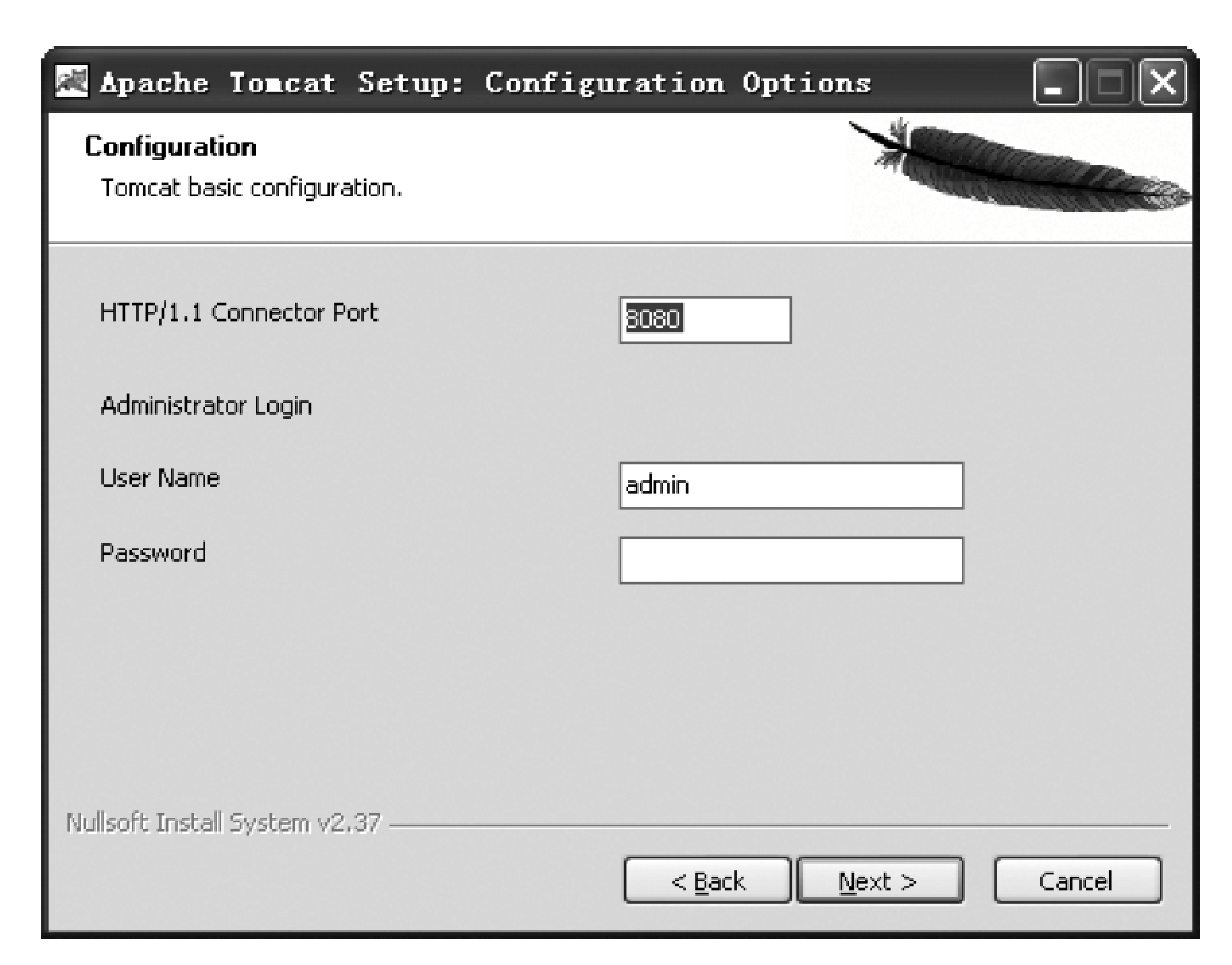


图 5-12　配置 Tomcat

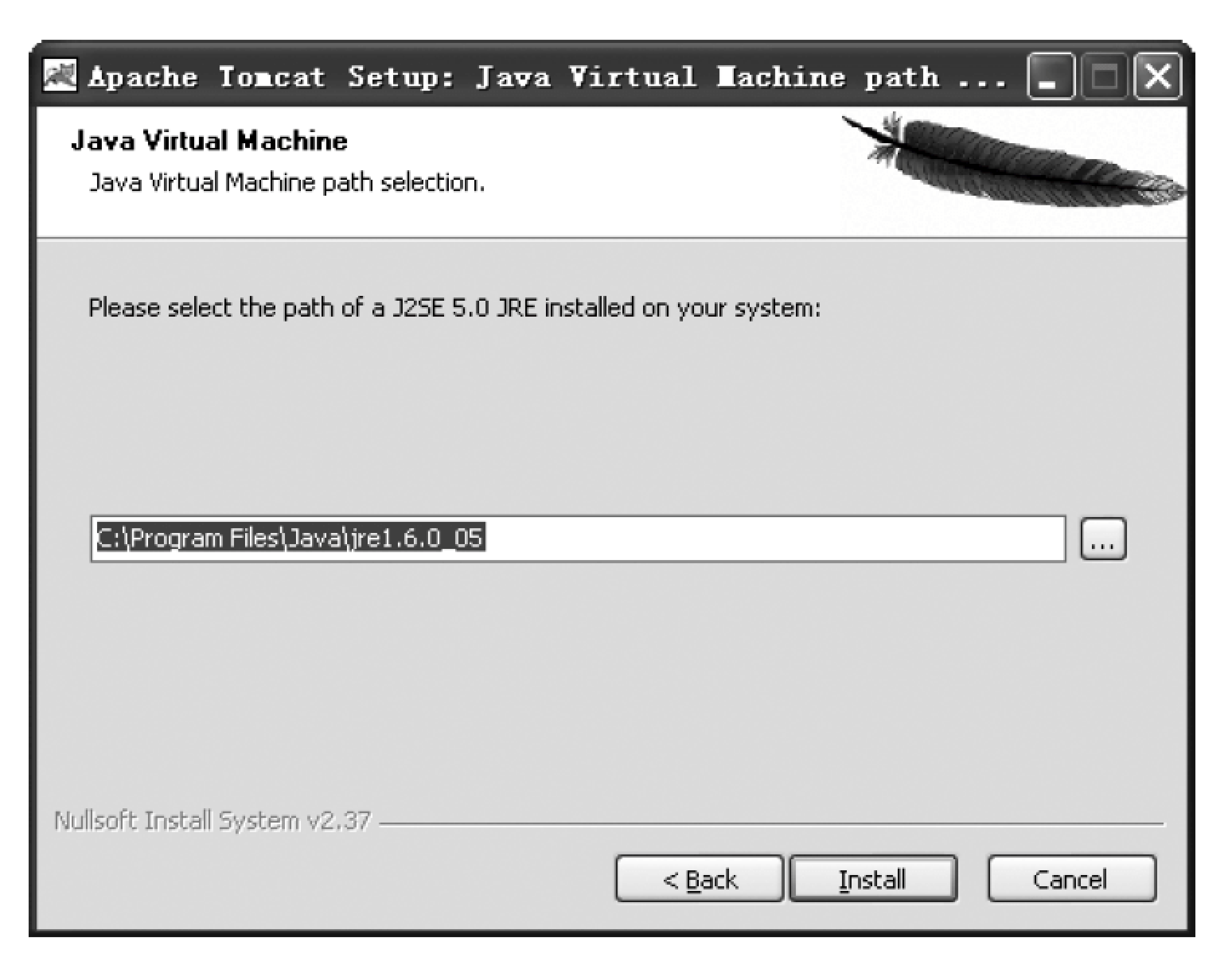


图 5-13　选择 Java 虚拟机的安装路径

(7) 成功安装后，程序会提示启动 Tomcat 并查看 readme 文档，如图 5-14 所示。Tomcat 正常启动后会在系统托盘区加载图标。右击该图标，可弹出一个快捷菜单，其中，【Stop Service】菜单项用于停止 Tomcat 服务，【Start Service】菜单项用于启动 Tomcat 服务。

(8) 至此，安装 Tomcat 的任务已经完成，请读者打开浏览器，在地址栏输入 http://localhost:8080，可以看到如图 5-15 所示的 Tomcat 的相关信息。



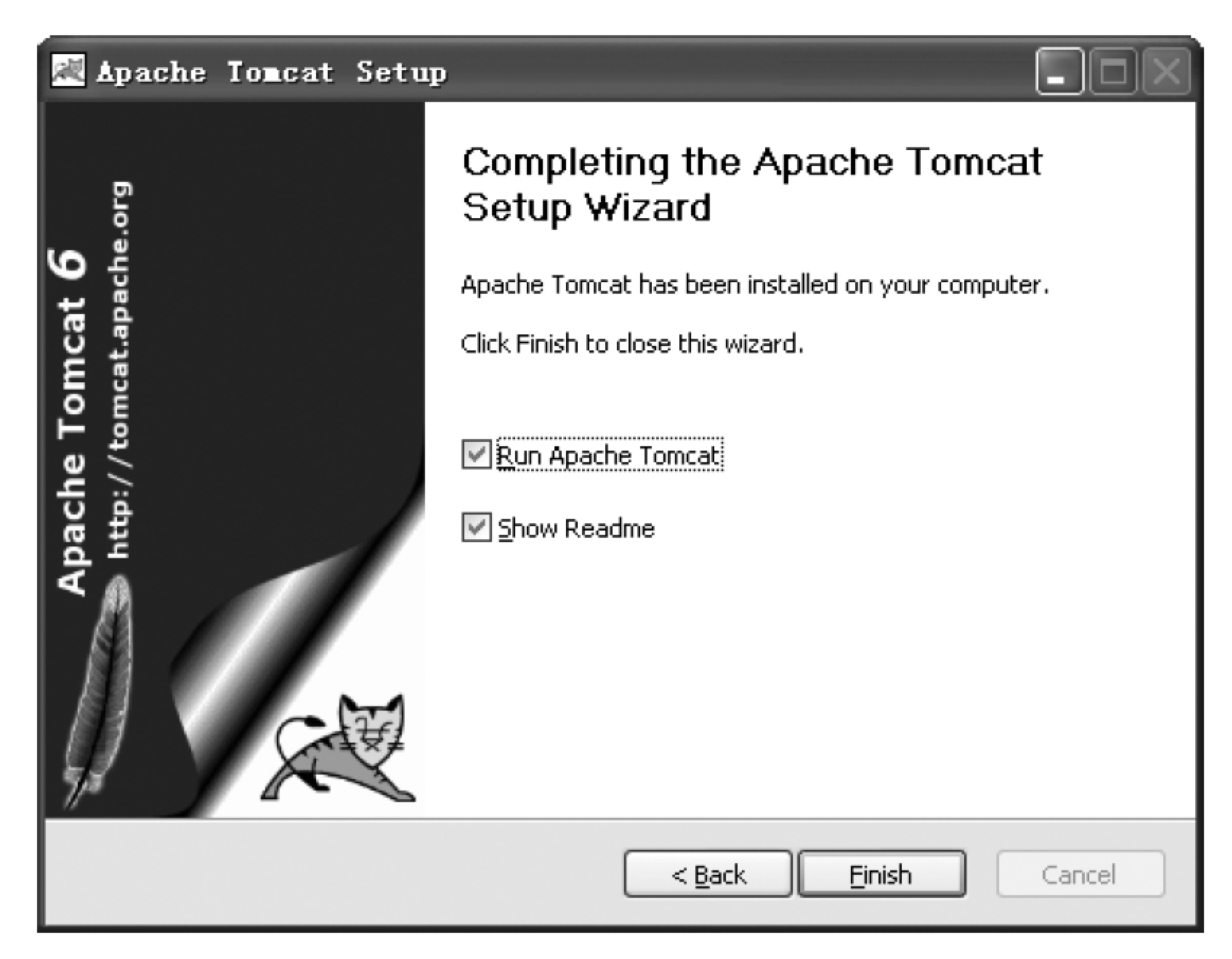


图 5-14　Tomcat 安装完成

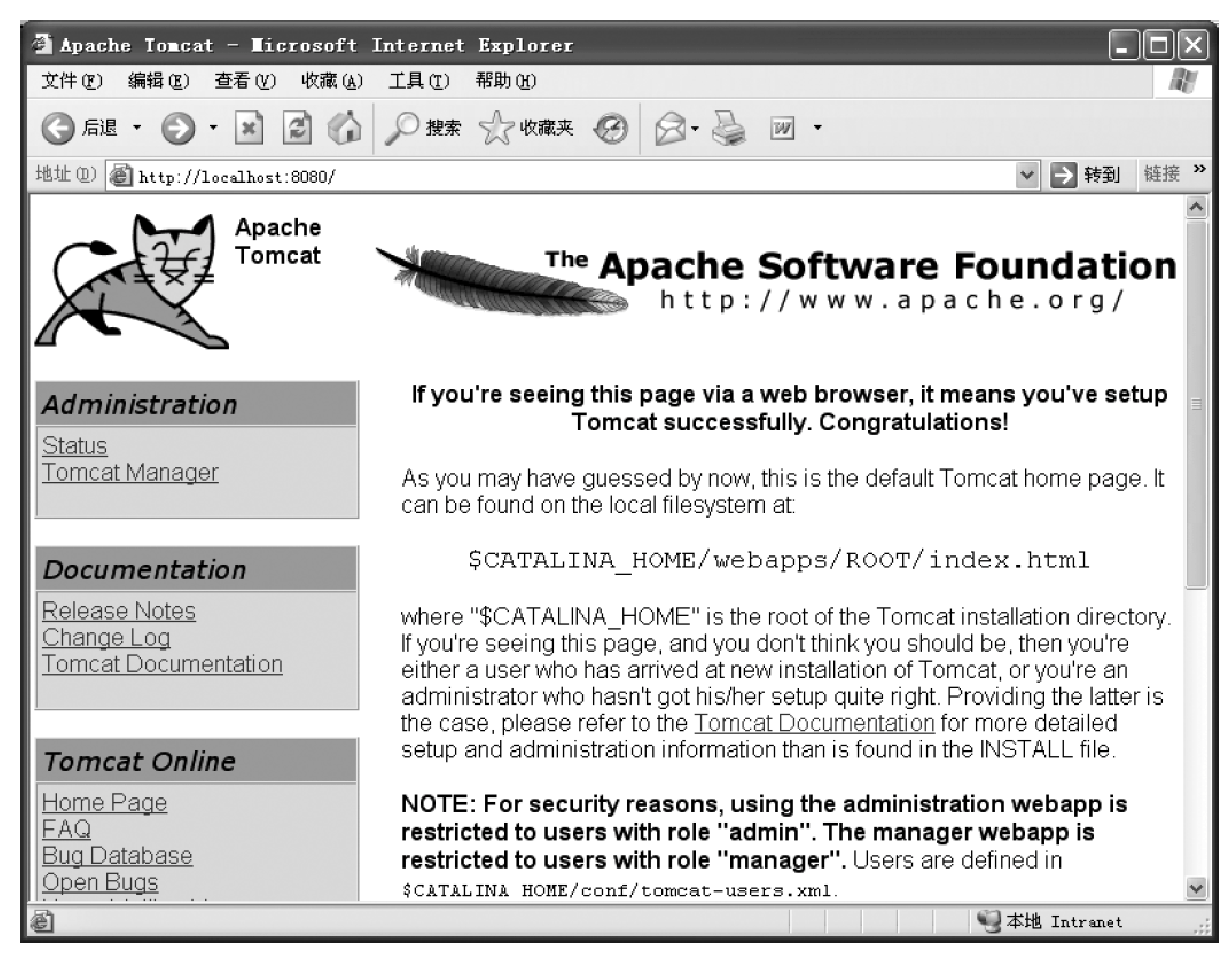


图 5-15　Tomcat 管理页面



5.2.3　相关知识拓展

1. Tomcat 简介

Tomcat 服务器是免费开放源代码的 Web 应用服务器。Tomcat 是 Apache 软件基金会(Apache Software Foundation)的 Jakarta 项目中的一个核心项目,由 Apache、Sun 和其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持,最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现,Tomcat 5 支持最新的 Servlet 2.4 和 JSP 2.0 规范。因为 Tomcat 技术先进、性能稳定,而且免费,因而深受 Java 爱好者的喜爱并得到了部分软件开发商的认可,成为目前比较流行的 Web 应用服务器。

Tomcat 运行时占用的系统资源小、扩展性好、支持负载平衡与邮件服务等开发应用系统常用的功能,而且它还在不断的改进和完善中,任何一个感兴趣的程序员都可以更改它或在其中加入新的功能。

Tomcat 是一个小型的轻量级应用服务器,在中小型系统和并发访问以及访问者不是很多的场合下被普遍使用,是开发和调试 JSP 程序的首选。若一台机器上配置好了 Apache 服务器,则可利用该机器响应对 HTML 页面的访问请求。实际上 Tomcat 是 Apache 服务器的扩展,但 Tomcat 是独立运行的,所以运行 Tomcat 时,它实际上是作为一个与 Apache 独立的进程单独运行的。

当配置正确时,Apache 为 HTML 页面服务,而 Tomcat 实际上运行 JSP 页面和 Servlet。另外,Tomcat 和 IIS、Apache 等 Web 服务器一样,具有处理 HTML 页面的功能,另外它还是一个 Servlet 和 JSP 容器,独立的 Servlet 容器是 Tomcat 的默认模式。不过,Tomcat 处理静态 HTML 的能力不如 Apache 服务器。

2. Tomcat 的配置

Tomcat 提供了一系列的配置文件,可以配置自己需要的 Tomcat 环境。Tomcat 的配置文件主要是基于 XML 的,如 server. xml、web. xml 等,只有 workers. properties 和 uriworkermap. properties 等几个少数文件是传统的配置文件。下面详细讨论 Tomcat 的主要配置文件。

(1) Tomcat 的主配置文件 server. xml

server. xml 文件描述了 Tomcat 的基本配置信息。server. xml 文件的各个元素及其属性的意义如表 5-1 所示。

表 5-1　server. xml 的元素

元　素　名	属　　性	说　　明
Server(该元素是 server. xml 文件最高级别的元素,它描述了一个 Tomcat 服务器)	port	port 指定一个端口,这个端口负责监听关闭 Tomcat 的请求
	shutdown	shutdown 指定向端口发送的命令字符串
Service	name	指定 Service 的名字



续表

元　素　名	属　　性	说　　明
Connector（表示客户端和Service之间的连接）	port	指定服务器端要创建的端口号，并在这个端口监听来自客户端的请求
	minThreads	服务器启动时创建的用于处理请求的最小线程数
	maxThreads	最多可以创建的处理请求的线程数
	enableLookups	如果为true，则可以通过调用request.getRemoteHost()进行DNS查询来得到远程客户端的实际主机名，若为false则不进行DNS查询，而是返回其IP地址
	redirectPort	指定服务器正在处理HTTP请求时，如果收到了一个SSL传输请求后，重定向端口号
	acceptCount	指定当所有可以使用的处理请求的线程数都被使用后，可以放到等待处理队列中的请求数，超过这个数的请求将不予处理
	connectionTimeout	以毫秒为单位，指定连接时的超时时间
Engine（表示指定Service中的请求处理机，接收和处理来自Connector的请求）	defaultHost	指定默认的处理请求的主机名，它至少应该与其中的一个host元素的name属性值是相同的
Context（每一个Context都描述了一个Tomcat的Web应用程序的目录）	docBase	Context的目录
	path	表示Context在Web服务器时的虚拟目录位置和目录名
	reloadable	如果属性为true，则Tomcat会自动检测应用程序的/WEB-INF/lib和/WEB-INF/classes目录的变化，自动装载新的应用程序，可以在不重启Tomcat的情况下改变应用程序
host（表示一个虚拟主机）	name	指定主机名
	appBase	应用程序基本目录，即存放应用程序的目录
	unpackWARs	如果为true，则Tomcat会自动将WAR文件解压，否则不解压，直接从WAR文件中运行应用程序
Logger（表示日志、调试和错误信息）	className	指定logger使用的类名，此类必须实现org.apache.catalina.Logger接口
	prefix	指定log文件的前缀
	suffix	指定log文件的后缀
	timestamp	如果为true，则log文件名中要加入时间，如下localhost_log.2001-10-04.txt
Realm（表示存放用户名、密码及role的数据库）	className	指定Realm使用的类名，此类必须实现org.apache.catalina.Realm接口
Valve（功能与Logger差不多，其prefix和suffix属性解释和Logger中的一样）	className	指定Valve元素所使用的类名，如果使用org.apache.catalina.valves.AccessLogValve类可以记录应用程序的访问信息



续表

元　素　名	属　　性	说　　明
	directory	指定 log 文件存放的位置
	pattern	有两个值，common 方式记录远程主机名或 IP 地址、用户名、日期、第一行请求的字符串、HTTP 响应代码、发送的字节数；combined 方式比 common 方式记录的值更多

下面使用 server. xml 文件设定一个新的 JSP 工作目录。设立新的 JSP 工作目录是比较简单的，只需要添加一个 Context 元素即可。例如，想要将在 D:\jspWorkDirectory 设定为一个新的 JSP 工作目录，并且使用户可以使用/myjsp 访问，只需要在 server. xml 文件中添加如下的 Context 元素：

```
<Context path="/myjsp" docBase="D: \jspWorkDirectory" reloadable="true" >
    </Context>
    </Host>
    </Engine>
    </Service>
</Server>
```

然后，就可在 D:\jspWorkDirectory 目录下创建 JSP 文件，在浏览器的地址栏中输入"http://Localhost: 8080/myjsp/ * . jps"，访问指定的 JSP 文件。

(2) 部署描述文件 web. xml

除 server. xml 文件外，另一个重要的配置文件为部署描述符文件 web. xml。所有部署描述符文件的顶层(根)元素都为 web-app。需要注意，XML 元素不像 HTML 标记，XML 元素是区分大小写的。因此，web-App 和 WEB-APP 都是不合法的，web-app 必须用小写。在 web-app 元素内，元素的次序也是很重要的。例如，servlet 元素必须出现在所有 servlet-mapping 元素之前。需要注意的是，所有这些元素都是可选的。

5.2.4　经验总结

本节讲述的是采用安装程序直接安装 Tomcat，不需要进行设置。而实际安装中还有一种情况，就是直接将程序拷入或解压到指定位置，然后进行相应的设置。具体的操作步骤如下：

(1) 安装 Tomcat 后，右击【我的电脑】，从弹出的菜单中选择【属性】|【高级】|【环境变量】|【新建】命令，添加以下环境变量(假定 Tomcat 安装在 C: \tomcat)：

```
CATALINA_HOME=C: \tomcat;
CATALINA_BASE=C: \tomcat;
```

如图 5-16 所示。

(2) 修改环境变量中的 classpath，把 Tomat 安装目录中的 common\lib 文件夹下的 servlet. jar 追加到 classpath 中去，修改后的 classpath 如下：



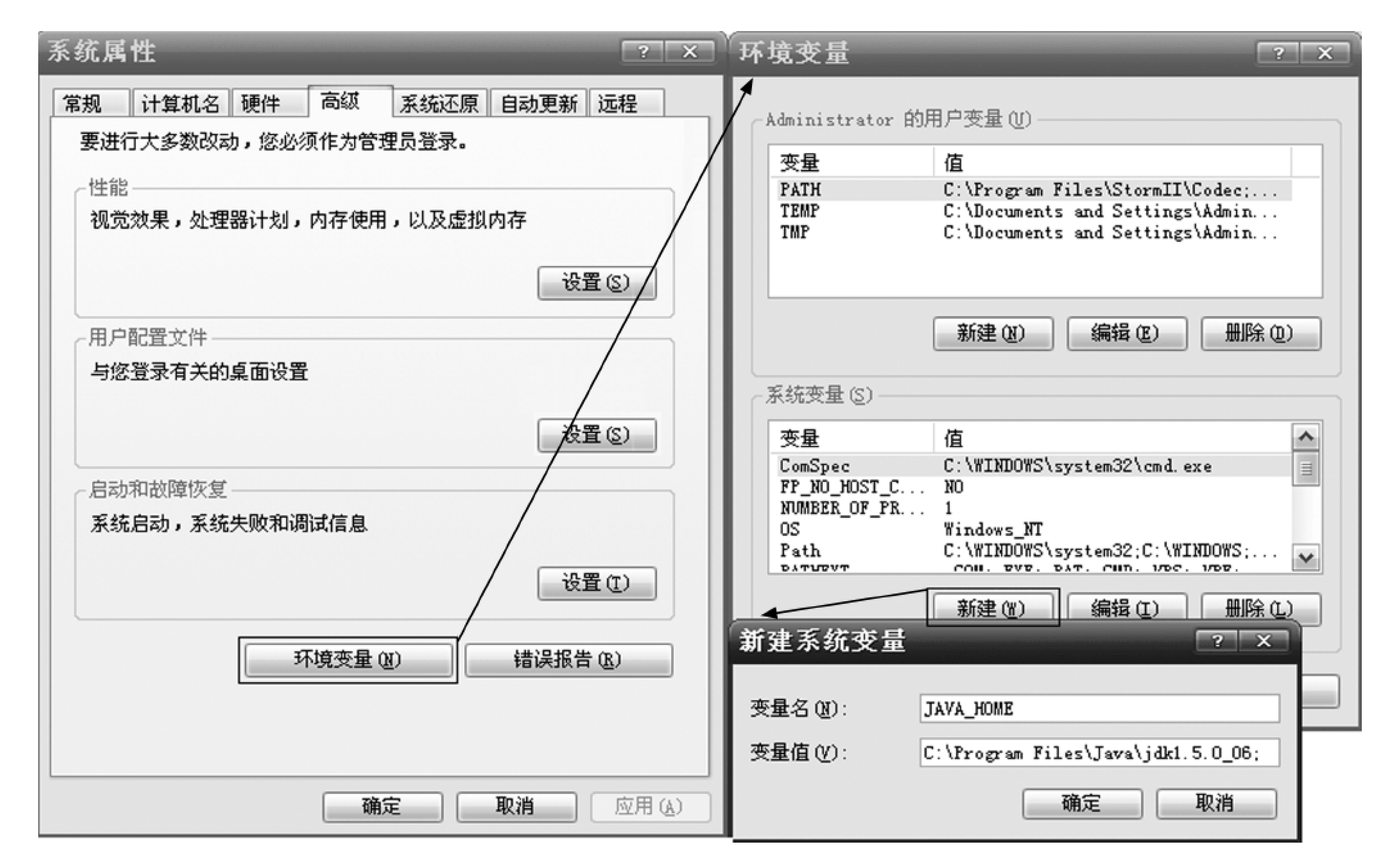


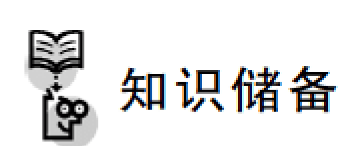
图 5-16　JAVA_HOME 变量的设置

```
classpath=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;%CATALINA_HOME%\common\
lib\servlet.jar;
```

（3）启动 Tomcat，在 IE 地址栏中输入 http://localhost:8080，如果看到 Tomcat 的欢迎页面，则说明安装成功了。

此处只是简单的步骤描述，若需要，可在互联网上搜索与 JDK 安装和 Tomcat 安装相关的文章，仔细阅读、参看，如，可搜索《搭建 Eclipse＋MyEclipse＋Tomcat 开发环境》来学习。

5.3　安装、配置 Microsoft SQL Server 2005



知识储备

越来越多的网站需要数据库系统的支持，常用的数据库系统有 Oracle（甲骨文）公司的 Oracle，Microsoft（微软）公司的 SQL Server，IBM 公司的 DB2，这些数据库系统各有特点，当网站在数据库方面没有特殊要求时，读者可以按照各自的熟练程度选择数据库系统。

安装 Microsoft SQL Server 2005 时，要根据需要选择合适的版本。Microsoft SQL Server 2005 有五个版本：企业版（Enterprise Edition）、标准版（Standard Edition）、工作组版（Workgroup Edition）、开发版（Developer Edition）和快递版（Express Edition），这些版本对安装环境有不同的要求，若安装环境不符合，可能会导致 SQL Server 2005 安装失败。

在这五个版本中，工作组版和快递版仅适合 32 位的平台，其他的版本均适合 32 位和 64 位的平台。对于操作系统的要求，除了企业版要求服务器操作系统外，其他的各版本均



可以在 WinXP Professional SP2 和 Win2000 Professional SP4 上安装。

SQL Server 2005 安装程序中带有一个系统配置检查器（SCC）程序，在安装时该程序将检查安装 SQL Server 2005 的计算机，并将每个检查项的状态与所要求的安装条件进行比较，给予相应的提示，对于不符合项则给出了解决相关问题的指导。读者首次安装 SQL Server 2005 时要注意 SCC 的检查结果，发现不符合项时需要解决，否则可能会影响 SQL Server 2005 的安装。

5.3.1　任务描述

构建“新闻类别网站”的开发环境之三：安装 Microsoft SQL Server 2005。

5.3.2　任务实现

具体操作步骤如下：

（1）若要开始安装过程，应将 SQL Server 2005 DVD 插入 DVD 驱动器。如果 DVD 驱动器的自动运行功能无法启动安装程序，应导航到 DVD 的根目录，然后启动 splash. hta。启动后出现安装导航界面，如图 5-17 所示，在此界面上选择【服务器组件、工具、联机丛书和示例】选项。



图 5-17　SQL Server 2005 安装导航界面

（2）出现如图 5-18 所示的【最终用户许可协议】页，阅读许可协议，再选中相应的复选框以接受许可条款和条件。接受许可协议后即可激活【下一步】按钮。若要继续，单击【下一步】按钮。若要结束安装程序，单击【取消】按钮。

（3）在如图 5-19 所示的【安装必备组件】界面上，安装程序将安装 SQL Server 2005 的必需软件。更新完成之后若要继续，单击【下一步】按钮。



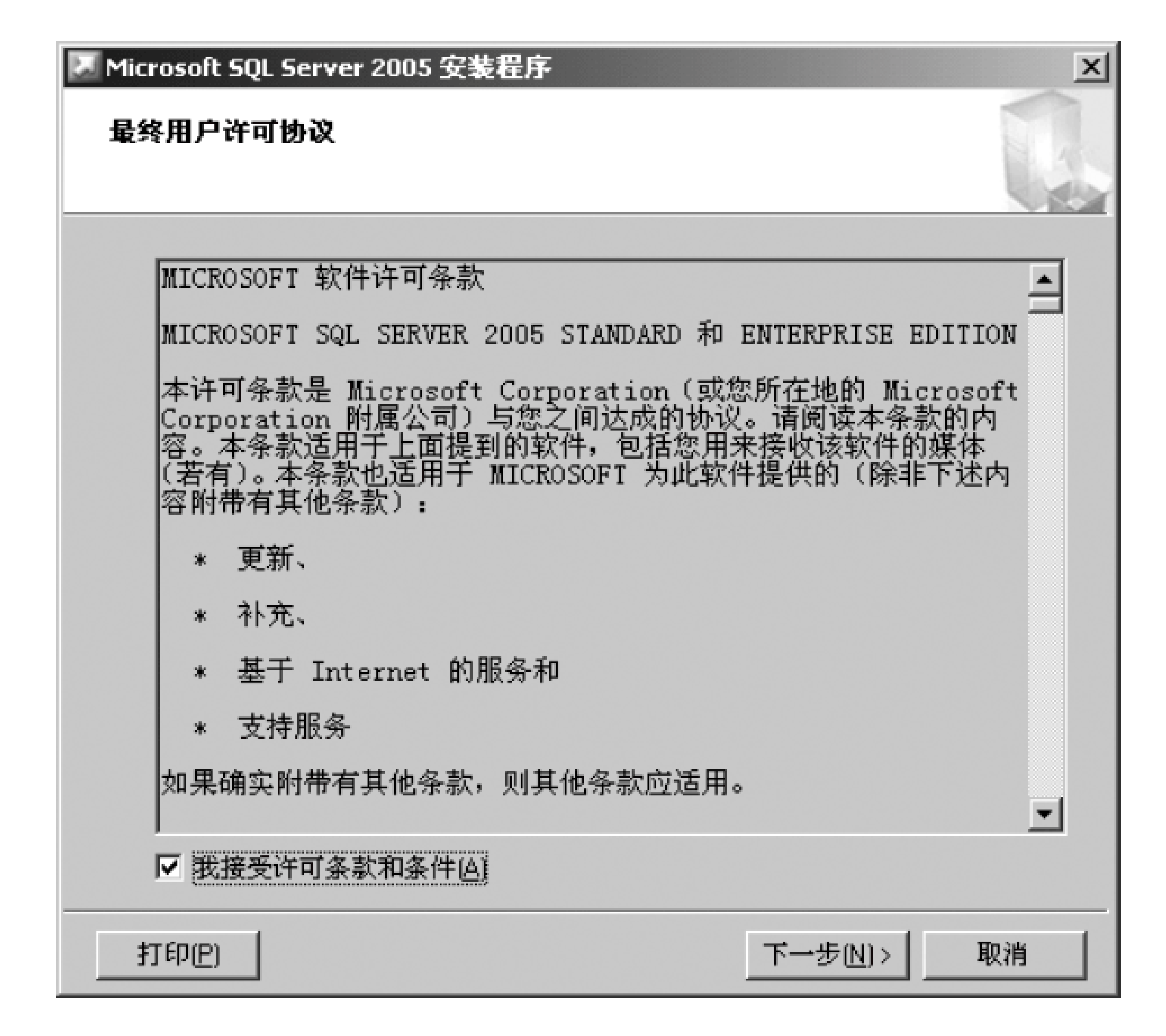


图 5-18　SQL Server 2005 最终用户许可协议界面

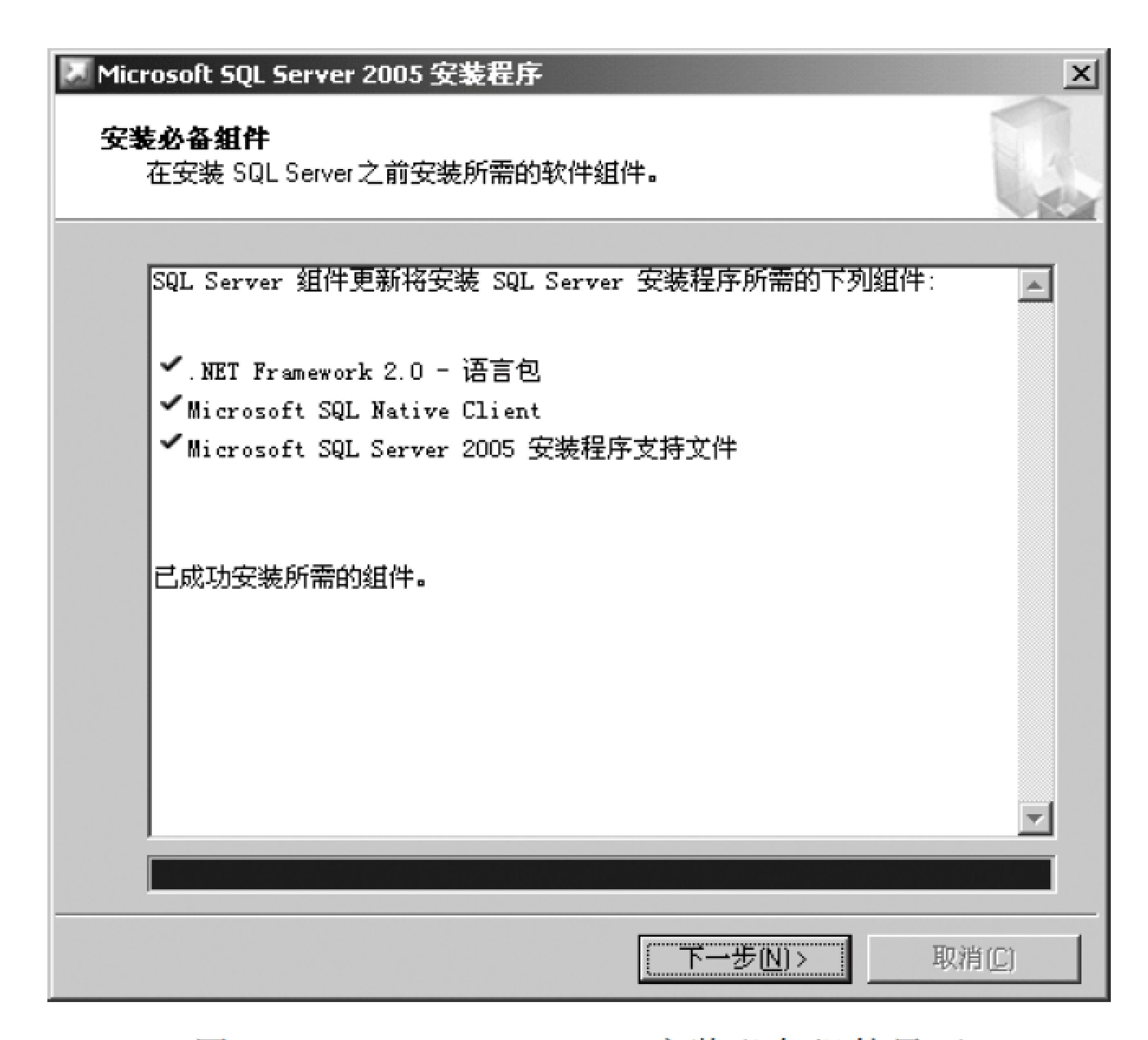


图 5-19　SQL Server 2005 安装必备组件界面

（4）在 SQL Server 安装向导的【欢迎】界面上，单击【下一步】按钮继续安装，界面如图 5-20所示。

（5）在图 5-21 所示的【系统配置检查】(SCC）界面上，将显示扫描安装 SQL Server 的计算机的系统配置结果，看看是否存在可能阻止安装程序运行的情况。有关配置检查项的信息，单击该页底部的【帮助】或参阅系统配置检查器的检查参数；若要中断扫描，单击【停止】按钮；若要显示按结果进行分组的检查项列表，单击【筛选器】按钮，然后从下拉列表中选择类别；若要查看 SCC 检查结果的报表，单击【报表】按钮，然后从下拉列表中选择选







图 5-20　SQL Server 2005 安装向导欢迎界面

项，选项包括查看报表、将报表保存到文件、将报表复制到剪贴板和以电子邮件形式发送报表。完成 SCC 扫描之后，单击【下一步】按钮。



图 5-21　SQL Server 2005 系统配置检查界面



（6）在如图 5-22 所示的【注册信息】界面上，在【姓名】和【公司】文本框中输入相应的信息，然后单击【下一步】按钮。

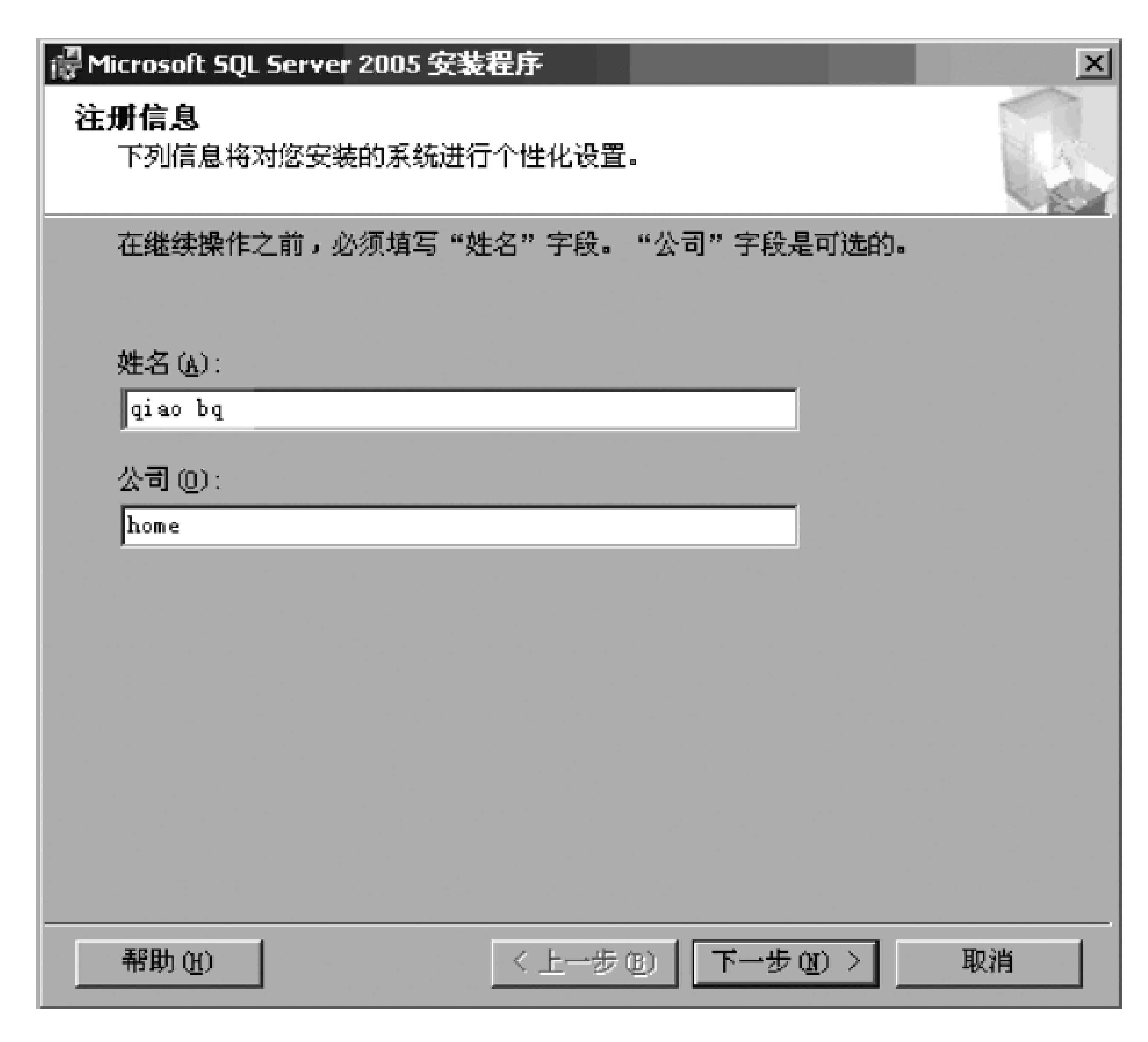


图 5-22　SQL Server 2005 系统注册信息界面

（7）在如图 5-23 所示的【要安装的组件】界面上，选择要安装的组件。其中前五个是服务器端组件，【SQL Server Database Services】是数据库服务组件，【Analysis Services】是分析服务组件，【Reporting Services】是报表服务组件，【Notification Services】是通知服务组件，【Integration Services】是集成服务组件，最后一个选项是【工作站组件、联机丛书和开发工具】。读者可以根据需要来选择要安装的组件，在“新闻类别网站”中选择第一个服务器组



图 5-23　SQL Server 2005 选择要安装的组件界面



件和最后一个工作站组件即可，单击【下一步】按钮继续安装。

（8）在如图 5-24 所示的【实例名】界面上，选择要安装默认实例还是特定实例（由安装者命名的实例）。每个 SQL Server 实例是由一组具有排列规则及其他特定设置选项的非重复的服务组成，其中包括目录结构、注册表结构和服务名称等信息。这些信息都保存在安装过程中所创建的特定实例的 ID 中。SQL Server 2005 支持在单个服务器上安装多个相互独立的 SQL Server 实例，但只有一个实例可以是默认实例，其他实例都必须是命名实例。默认实例名在连接时不需要客户端指定实例名称即可建立连接，而命名实例则需要指定实例名才能建立连接。所安装的计算机上没有默认实例时，才可以安装默认实例。若要安装新的命名实例，单击【命名实例】，然后在提供的空白处键入一个唯一的实例名，界面如图 2-8所示。如果已经安装了默认实例或已命名实例，并且为安装的软件选择了现有实例，那么安装程序将升级所选实例并提供安装其他组件的选项。首次安装选择默认实例即可。设置完成后单击【下一步】按钮。

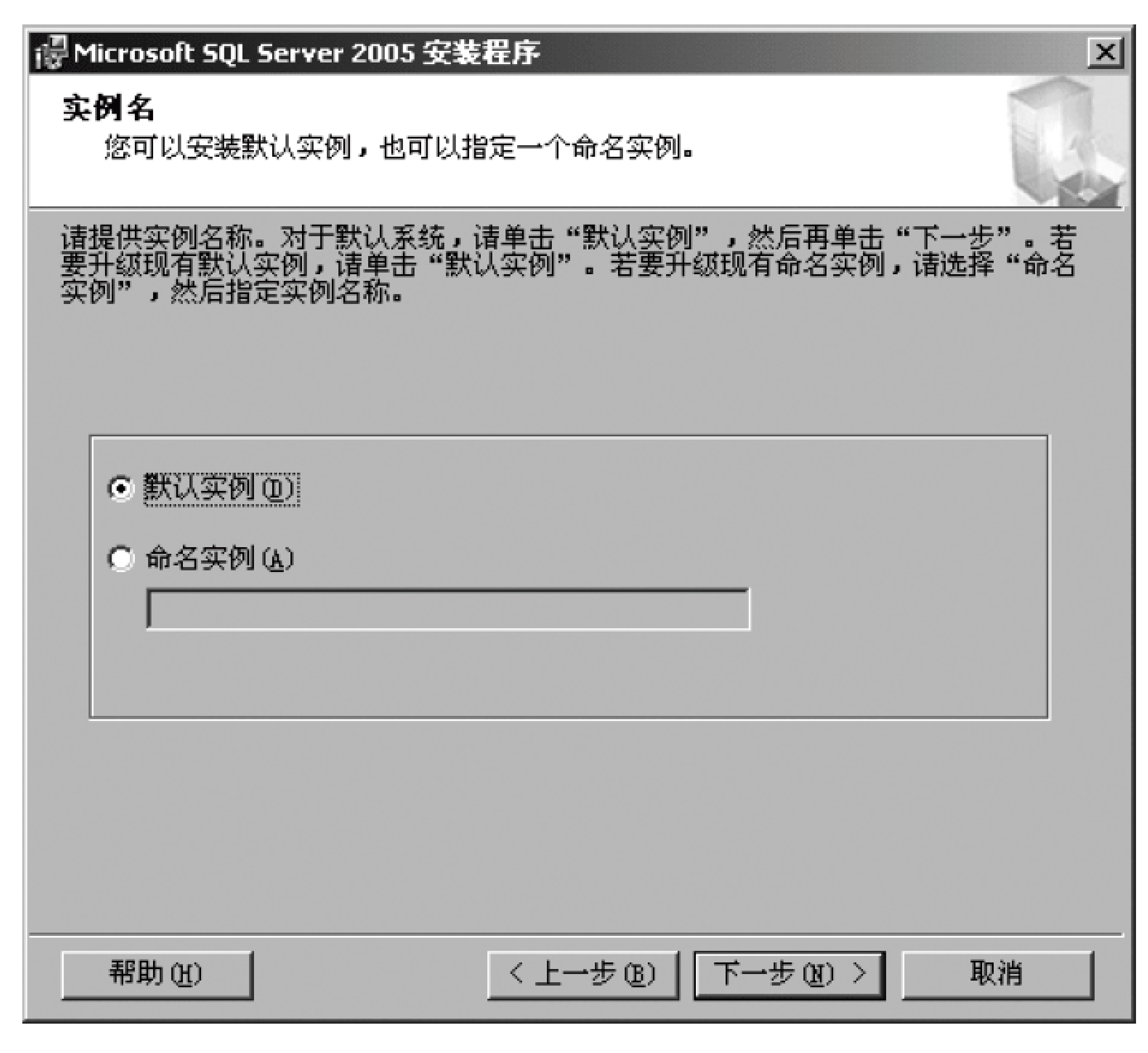


图 5-24　SQL Server 2005 指定或命名实例名界面

（9）在如图 5-25 中所示的【服务账户】界面上，为 SQL Server 服务账户指定用户名、密码和域名。根据需要可以让所有服务都使用一个账户，也可以为各个服务指定单独的账户。若要为各个服务指定单独的账户，请选中【为每个服务账户进行自定义】，从下拉框中选择服务名称，然后为该服务设定登录密码，此处选择【使用内置系统账户】。继续安装，单击【下一步】按钮。

注意

域名不能为完整的域名系统（DNS）名称。例如，如果 DNS 名称为 my-domain-name.com，应在【域】字段中使用 my-domain-name。在 SQL Server 中，【域】字段不接受 my-domain-name.com。





图 5-25　为 SQL Server 2005 选定服务账户界面

(10) 在如图 5-26 所示的【身份验证模式】界面上，选择要用于 SQL Server 安装的身份验证模式。采用混合模式验证时，必须输入并确认用于 sa 登录的密码。微软的 SQL Server 2005 联机丛书中建议最好使用 Windows 身份验证。这里的身份验证模式设置在 SQL Server 的使用中可以根据需要再做修改。对于 sa 账户，为了提高系统的安全性，已经不再像 SQL Server 2000 那样支持空密码了。设置完成后单击【下一步】按钮。



图 5-26　SQL Server 2005 身份验证模式界面



在“新闻类别网站”的应用中，建议读者选择【使用 Windows 身份验证和 SQL Server 身份验证】。

(11) 在如图 5-27 所示的【排序规则设置】界面上，指定 SQL Server 实例的排序规则。可以按照默认设置不做修改。单击【下一步】按钮。



图 5-27　SQL Server 2005 排序规则设置界面

(12) 在如图 5-28 所示的【准备安装】界面上，查看要安装的 SQL Server 功能和组件的摘要。若要继续安装，请单击【安装】按钮。



图 5-28　SQL Server 2005 的准备安装界面



(13) 在如图 5-29 所示的【安装进度】界面上,可以监视安装过程中的安装进度。若要在安装期间查看组件的日志文件,应在【安装进度】界面上单击产品或状态名称。



图 5-29　SQL Server 2005 的安装进度界面

(14) 在如图 5-30 所示的【完成 Microsoft SQL Server 2005 安装】界面上,可以通过单击此界面上提供的链接查看安装摘要日志。单击【完成】按钮可结束安装,退出 SQL Server 安装向导。

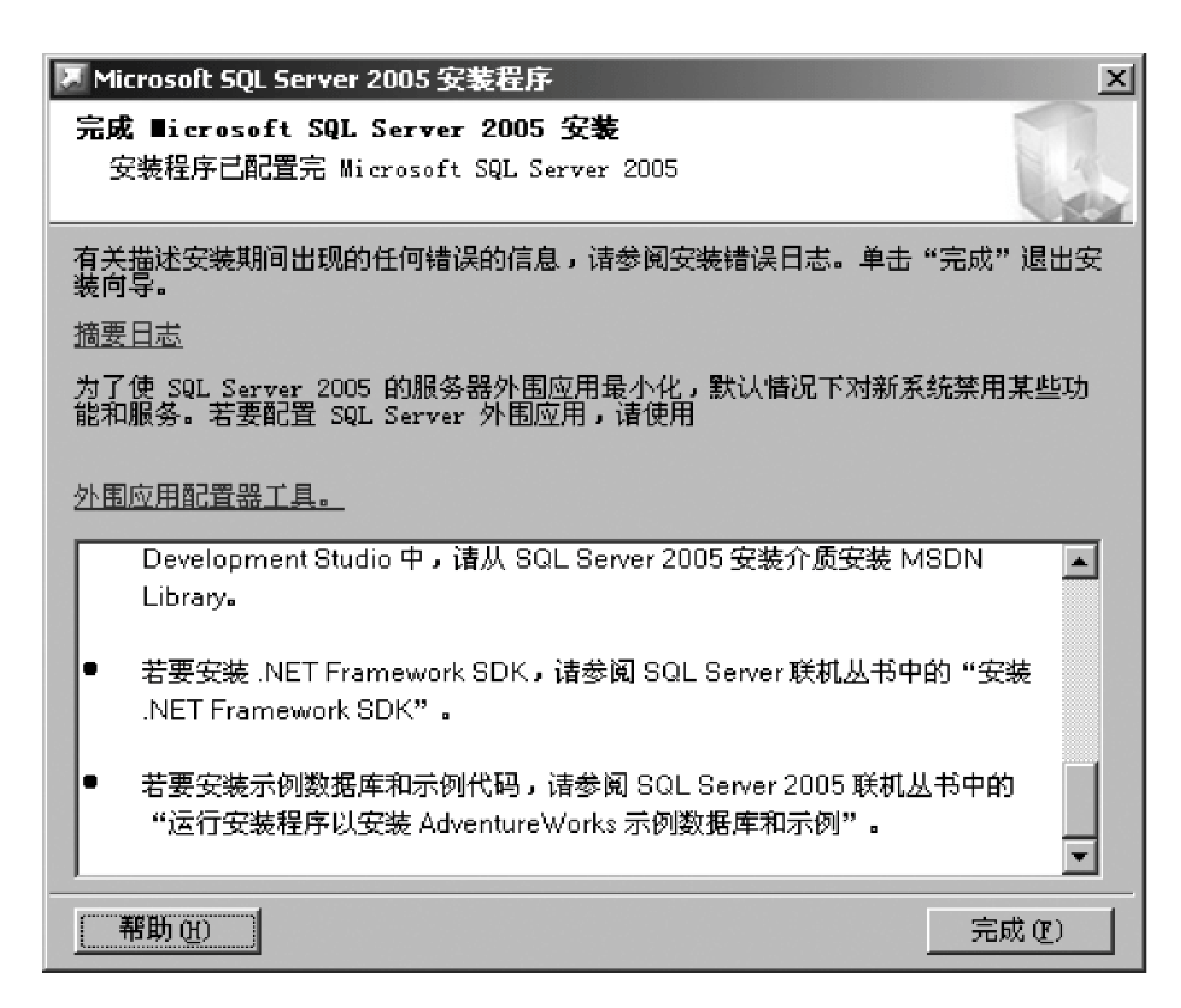


图 5-30　SQL Server 2005 的安装成功界面

(15) 如果得到重新启动计算机的指示,应立即进行操作。完成安装后,阅读来自安装程序的消息是很重要的。如果未能重新启动计算机,可能会导致以后运行安装程序的失败。



5.5.3　相关知识拓展

1. Microsoft SQL Server 2005 各版本简介

- SQL Server 2005 Enterprise Edition：适合运行在 32 位平台和 64 位平台上。Enterprise Edition 达到了支持超大型企业进行联机事务处理（OLTP）、高度复杂的数据分析、数据仓库系统和网站所需的性能水平。Enterprise Edition 的全面商业智能和分析能力及其高可用性功能（如故障转移群集），使它可以处理大多数关键业务的企业工作负荷。Enterprise Edition 是最全面的 SQL Server 版本，是超大型企业的理想选择，能够满足最复杂的要求。该版本还推出了一种适用于 32 位或 64 位平台的 120 天的 Evaluation Edition。
- SQL Server 2005 Standard Edition：适合运行在 32 位平台和 64 位平台上。Standard Edition 是适合中小型企业的数据管理和分析平台。它包括电子商务、数据仓库和业务流解决方案所需的基本功能。Standard Edition 的集成商业智能和高可用性功能可以为企业提供支持其运营所需的基本功能。Standard Edition 是需要全面的数据管理和分析平台的中小型企业的理想选择。
- SQL Server 2005 Workgroup Edition：仅适合运行在 32 位平台上。Workgroup Edition 是适合小型企业的数据管理和分析平台，能提供在数据库大小和用户数量上没有限制的数据管理解决方案。Workgroup Edition 可以用做前端 Web 服务器，也可以用于部门或分支机构的运营。它包括 SQL Server 产品系列的核心数据库功能，并且可以轻松地升级至 Standard Edition 或 Enterprise Edition。Workgroup Edition 是理想的入门级数据库，具有可靠、功能强大且易于管理的特点。
- SQL Server 2005 Developer Edition：适合运行在 32 位平台和 64 位平台上。Developer Edition 可以在 SQL Server 上生成任何类型的应用程序，它包括 SQL Server 2005 Enterprise Edition 的所有功能，但有许可限制，只能用于开发和测试系统，而不能用做生产服务器。Developer Edition 是独立软件供应商（ISV）、咨询人员、系统集成商、解决方案供应商以及创建和测试应用程序的企业开发人员的理想选择。Developer Edition 可以根据生产需要升级至 SQL Server 2005 Enterprise Edition。
- SQL Server 2005 Express Edition：仅适合运行在 32 位平台上。SQL Server Express 是一个免费、易用且便于管理的数据库。SQL Server Express 与 Microsoft Visual Studio 2005 集成在一起，可以轻松开发功能丰富、存储安全、可快速部署的数据驱动应用程序。SQL Server Express 是免费的，可以再分发（受制于协议），还可以起到客户端数据库以及基本服务器数据库的作用。SQL Server Express 是低端 ISV、低端服务器用户、创建 Web 应用程序的非专业开发人员以及创建客户端应用程序的编程爱好者的理想选择。



2．Microsoft SQL Server 2005 的组件介绍

Microsoft SQL Server 2005 的运行模式是 C/S 模式，其组件分为两大类，一类是在服务器端运行的服务器组件；另一类是在客户端运行的客户端组件。在前面的安装过程中，(见图 5-21)，我们建议读者在本地机上安装了服务器组件中的 SQL Server Database Services(数据库服务组件)和工作站组件(客户端)。下面对各组件作进一步的介绍，扩充大家的知识面。

(1) 服务器端组件

- SQL Server 数据库服务组件(SQL Server Database Services)：提供用于存储、处理和保护数据的核心服务，提供了受控访问和快速事务处理，以满足企业内最苛刻的数据消费应用程序的要求。数据库服务组件还提供了大量的支持以保持高可用性。通过数据库服务组件可以完成的任务包括：设计并创建数据库以保存系统所需的关系表或 XML 文档；实现系统以访问和更改数据库中存储的数据，包括实现网站或使用数据的应用程序，以及生成使用 SQL Server 工具和实用工具以使用数据的过程；为单位或客户部署实现的系统；提供日常管理支持以优化数据库的性能。
- 分析服务组件(Analysis Services)：用于创建复杂的联机分析处理 (OLAP) 和数据挖掘解决方案。Analysis Services 工具提供了设计、创建和管理来自数据仓库的多维数据集和数据挖掘模型的功能，还提供对 OLAP 数据和数据挖掘数据的客户端访问。
- 报表服务组件(Reporting Services)：用于生成从多种关系数据源和多维数据源提取内容的企业报表，发布以各种格式查看的报表，以及集中管理安全性和订阅。生成的报表可以通过基于 Web 的连接进行查看，也可以作为 Microsoft Windows 应用程序的一部分进行查看。报表服务组件的安装和使用需要 Internet 信息服务 (IIS) 5.0 或更高的版本支持，并需要 Microsoft Internet Explorer 6.0 Service Pack (SP) 1 的版本支持。
- 通知服务组件(Notification Services)：通知服务组件是一个应用程序开发和部署平台，其应用程序可以生成通知并将通知发送给订阅方或各种设备。使用 Notification Services 可以快速创建和部署应用程序，使应用程序在需要时能够支持数百万个订阅方。
- 集成服务组件(Integration Services)：用于创建执行提取、转换和加载数据(ETL)处理的可靠而复杂的解决方案。它包括用于生成和调试包的图形工具和向导；用于执行工作流函数(如 FTP 操作)、执行 SQL 语句或发送电子邮件的任务；用于提取和加载数据的数据源和目标；用于清理、聚合、合并和复制数据的转换；用于管理集成服务组件的管理服务；用于对集成服务组件对象模型编程的应用程序编程接口(API)。

(2) 客户端组件

客户端组件主要包括连接组件、管理工具、Business Intelligence Development Studio 和 SQL Server 联机丛书。

- 连接组件：客户端最主要的组件就是连接组件，它主要安装用于客户端和服务器之间通信的组件，以及用于 DB-Library、ODBC 和 OLE DB 的网络库。连接组件实现



的功能主要是通过 SQL Server 配置管理器来实现的,包括管理与 SQL Server 相关联的服务、配置 SQL Server 使用的网络协议以及在 SQL Server 客户端的计算机上管理网络连接配置。可以单击打开【程序】|【Microsoft SQL Server 2005】|【配置工具】|【SQL Server 配置管理器】来配置连接组件。如图 5-31 所示是【SQL Server 网络配置】|【MSSQLSERVER 的协议】的配置界面。如图 5-32 所示是【SQL Native Client】|【客户端协议】配置界面。

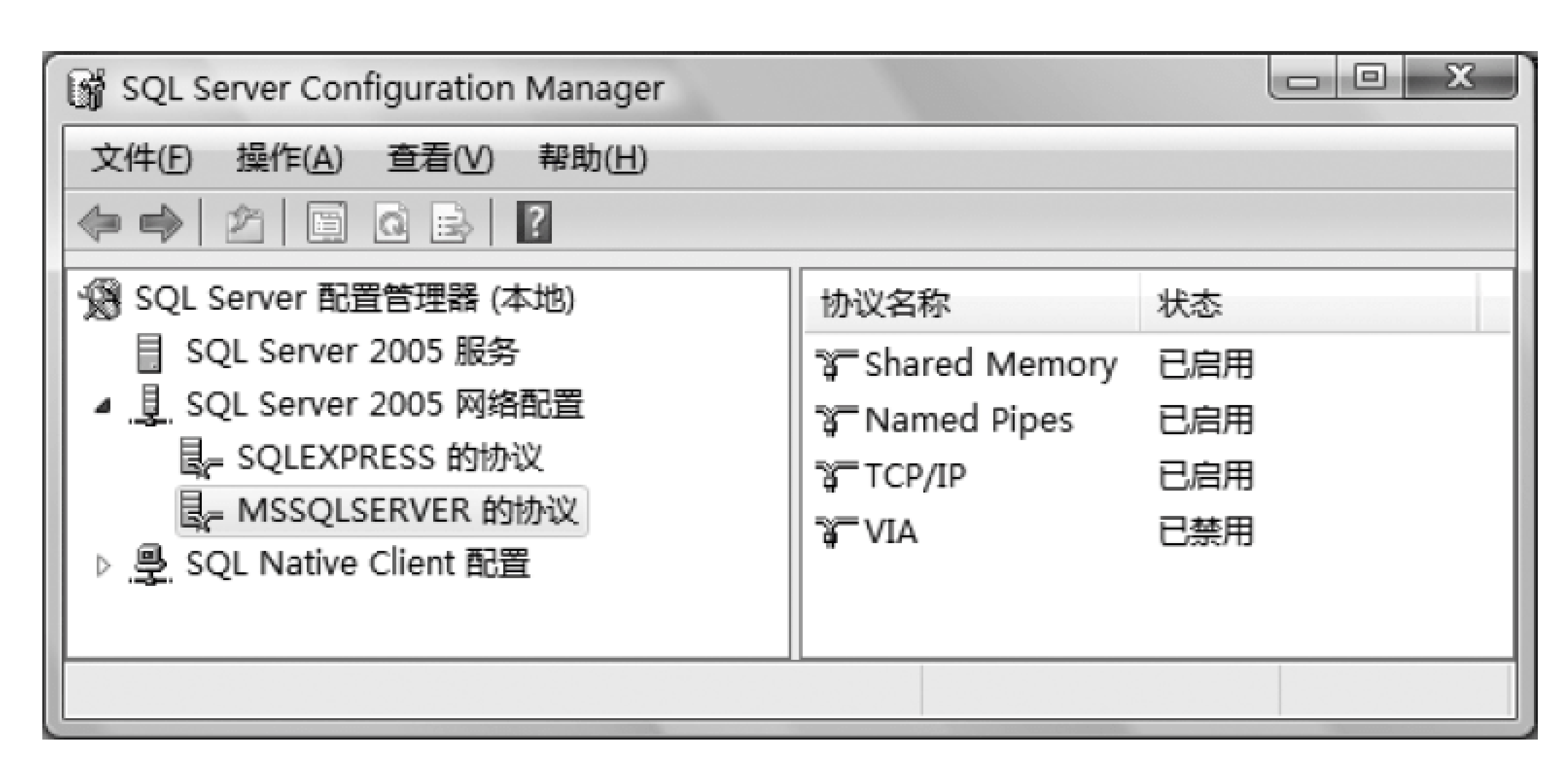


图 5-31　MSSQLSERVER 的协议配置

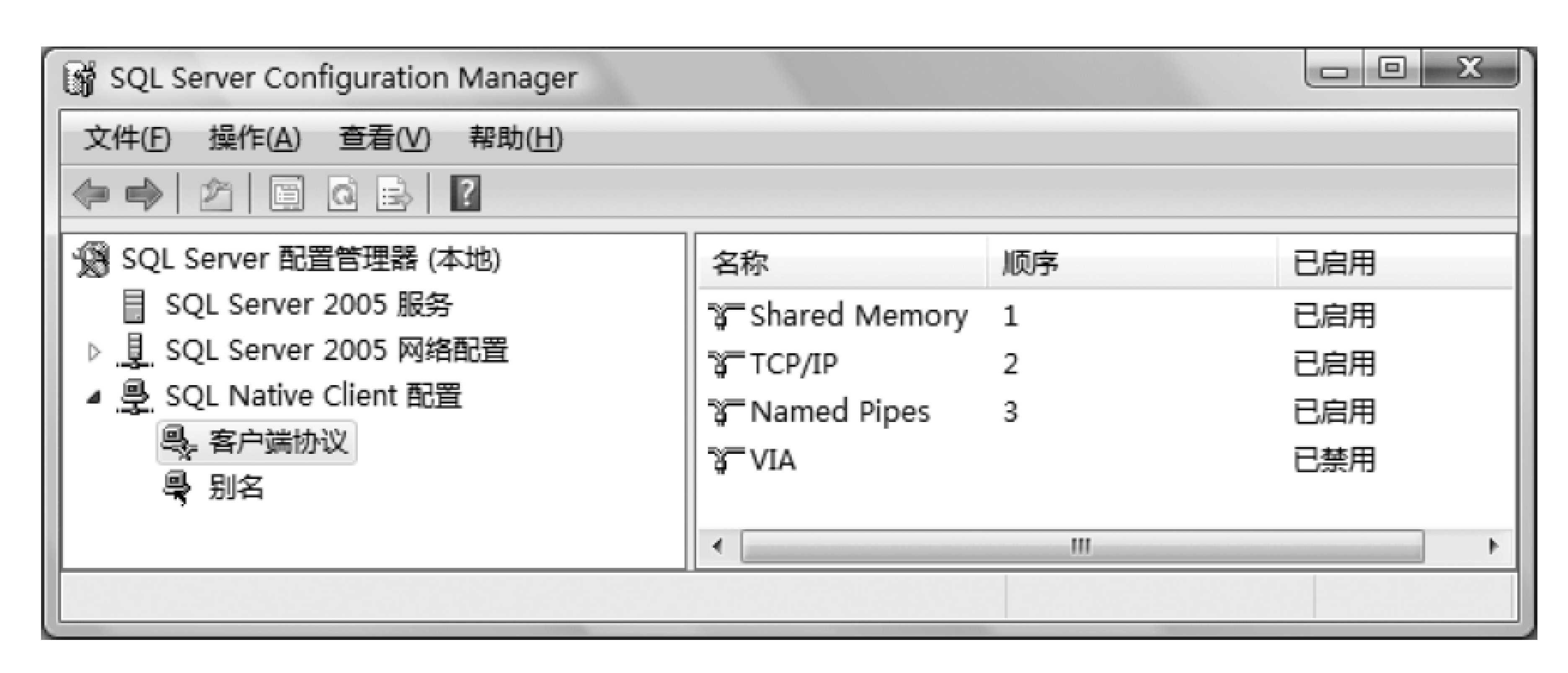


图 5-32　客户端的协议配置

- 管理工具:Microsoft SQL Server Management Studio 是 Microsoft SQL Server 2005 提供的一种新集成环境,用于访问、配置、控制、管理和开发 SQL Server 的所有组件。SQL Server Management Studio 将一组多样化的图形工具与多种功能齐全的脚本编辑器组合在一起,可为各种技术级别的开发人员和管理员提供对 SQL Server 的访问。这个工具将以前版本的 SQL Server 中所包括的企业管理器、查询分析器和 Analysis Manager 功能整合到单一环境中。
- 开发工具:Business Intelligence Development Studio 是用于开发包括 Analysis Services、Integration Services 和 Reporting Services 项目在内的商业解决方案的主要环境。每个项目类型都提供了用于创建商业智能解决方案所需对象的模板,并提供了用于处理这些对象的各种设计器、工具和向导。安装这个软件开发工具包需要 Internet Explorer 6.0 SP1 的支持。



- 帮助工具：SQL Server 联机丛书是 SQL Server 2005 的核心文档，从中可以获得大量的使用帮助和 SQL Server 2005 的相关知识。SQL Server 2005 教程可以帮助用户了解和学习使用 SQL Server 2005 提供的各种实用工具。SQL Server 2005 通过示例演示提供数据库引擎、Analysis Services、Reporting Services 和 Integration Services 的示例代码和示例应用程序，通过对示例的了解可以很快地学会 SQL Server 技术。

3. 启动和停止 SQL Server 服务

启动、停止 Microsoft SQL Server 2005 服务的方法不止一种，读者可以参考下面的方法来操作。

（1）使用外围配置应用器启动、停止 SQL Server 实例

在【开始】菜单中，依次选择【程序】|【Microsoft SQL Server 2005】|【配置工具】，然后单击【SQL Server 外围应用配置器】，启动的界面如图 5-33 所示。

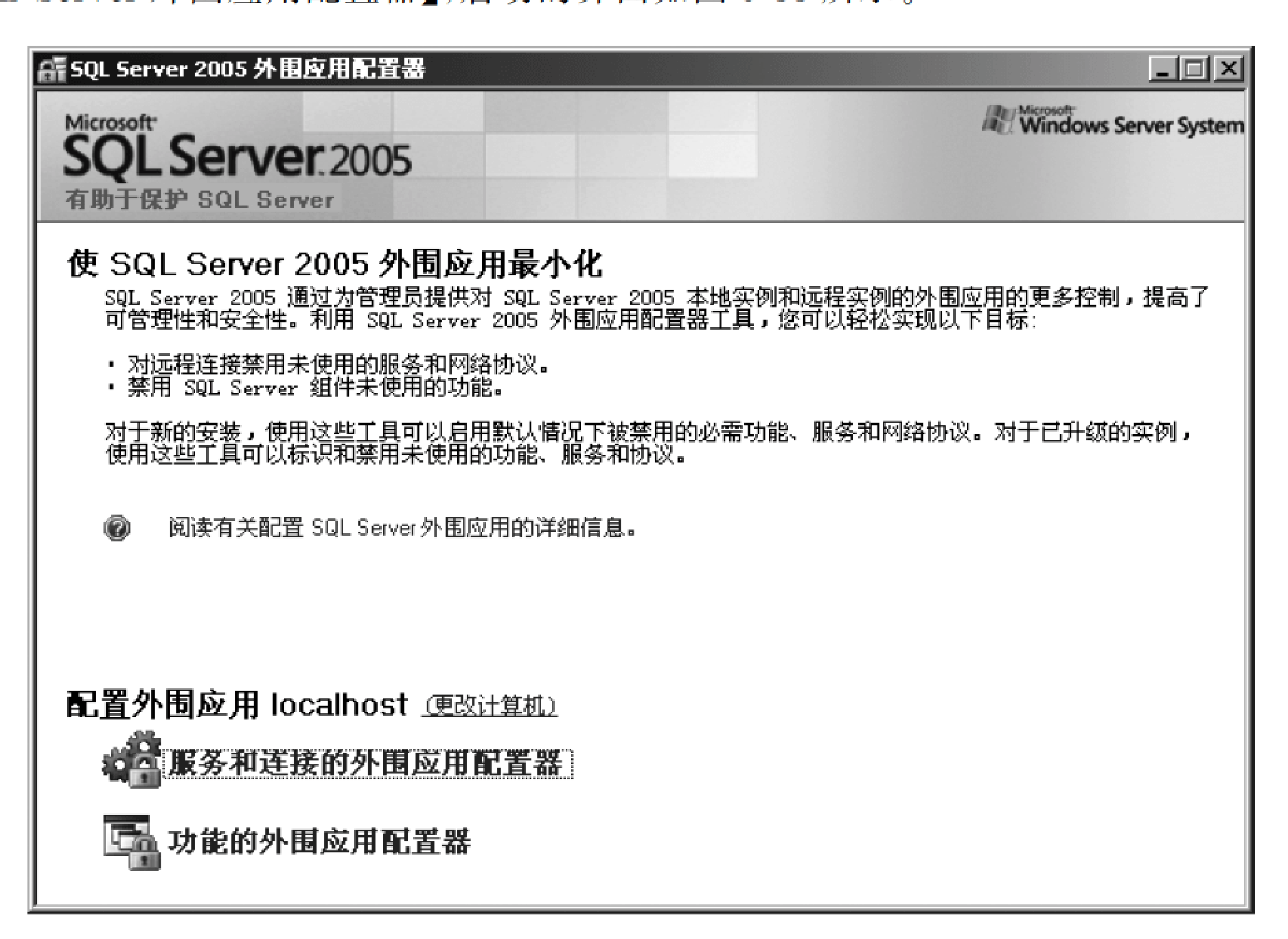


图 5-33　外围应用配置器界面

在外围应用配置器界面中，单击【服务和连接的外围应用配置器】，在出现的【服务和连接的外围应用配置器】界面中，展开【MSSQLSERVER】|【Database Engine】|【服务】，单击【启动】按钮，即可启动 MSSQLSERVER 实例上的 Database Engine 服务，如图 5-34 所示；单击【停止】按钮，则可停止该服务。

（2）使用 SQL Server Configuration Manager 启动 SQL Server 实例

在【开始】菜单中，依次指向【所有程序】|【Microsoft SQL Server 2005】|【配置工具】，然后单击【SQL Server Configuration Manager】，即可启动如图 5-35 所示的 SQL Server 配置管理器。



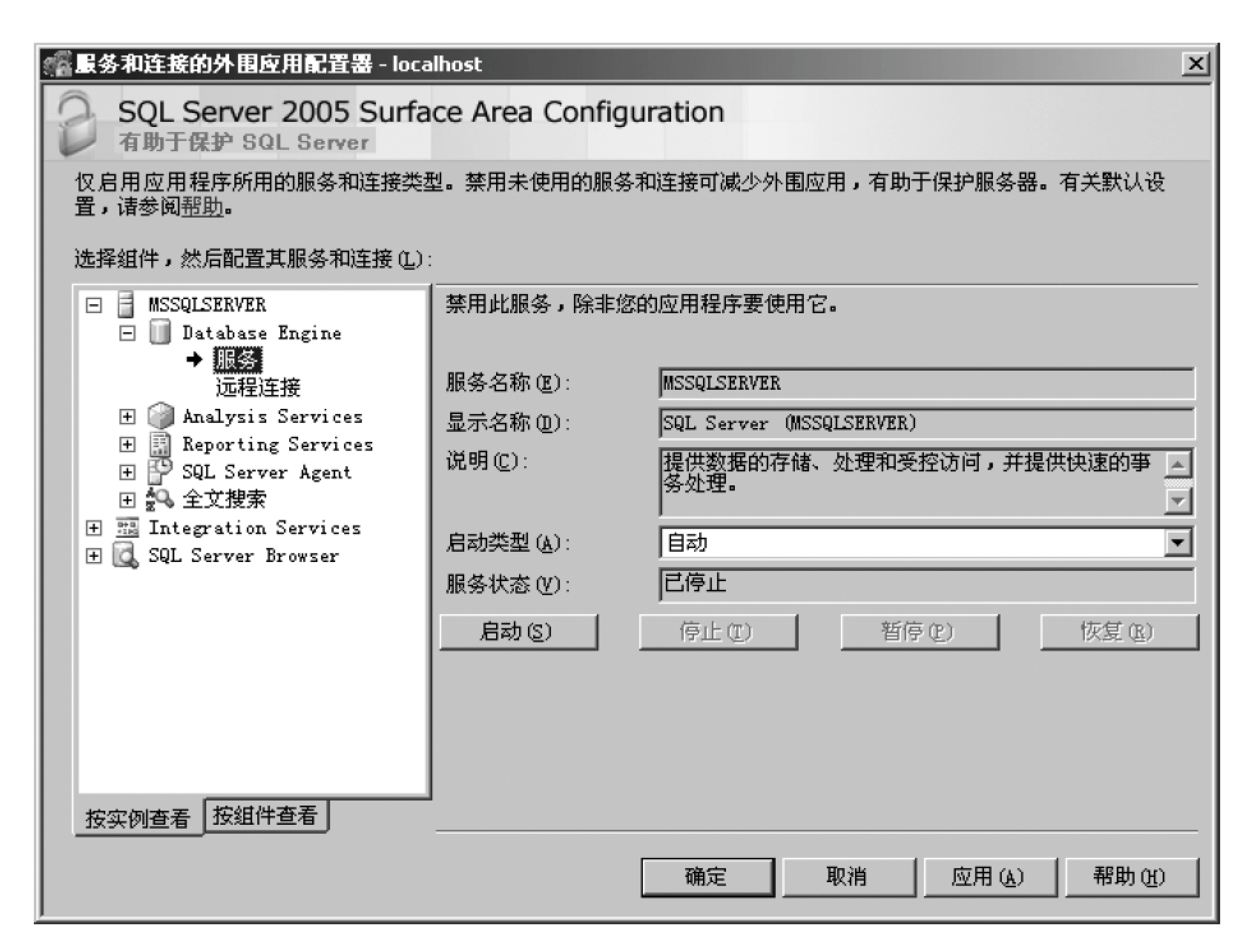


图 5-34　启动 Database Engine 服务的界面



图 5-35　SQL Server 配置管理器

在 SQL Server 配置管理器中，单击【SQL Server 2005 服务】，再单击 SQL Server (MSSQLSERVER)。然后通过单击工具栏的按钮暂停该服务，单击按钮停止该服务，单击按钮重启该服务，单击按钮则可以启动该服务。

4. 初识 Management Studio

Management Studio 是 SQL Server 最主要的客户端组件，对数据库的数据访问操作需要通过它来完成。下面简单介绍它的启动方法。

(1) 在【开始】菜单上，选择【所有程序】|【Microsoft SQL Server 2005】，再单击【SQL Server Management Studio】。



(2) 在如图 5-36 所示的【连接到服务器】对话框中，单击【连接】按钮，连接到 SQL Server 服务器。



图 5-36　连接到服务器

(3) 与服务器连接后的 SQL Server Management Studio 是一个功能强大且灵活的客户端操作工具，界面如图 5-37 所示。

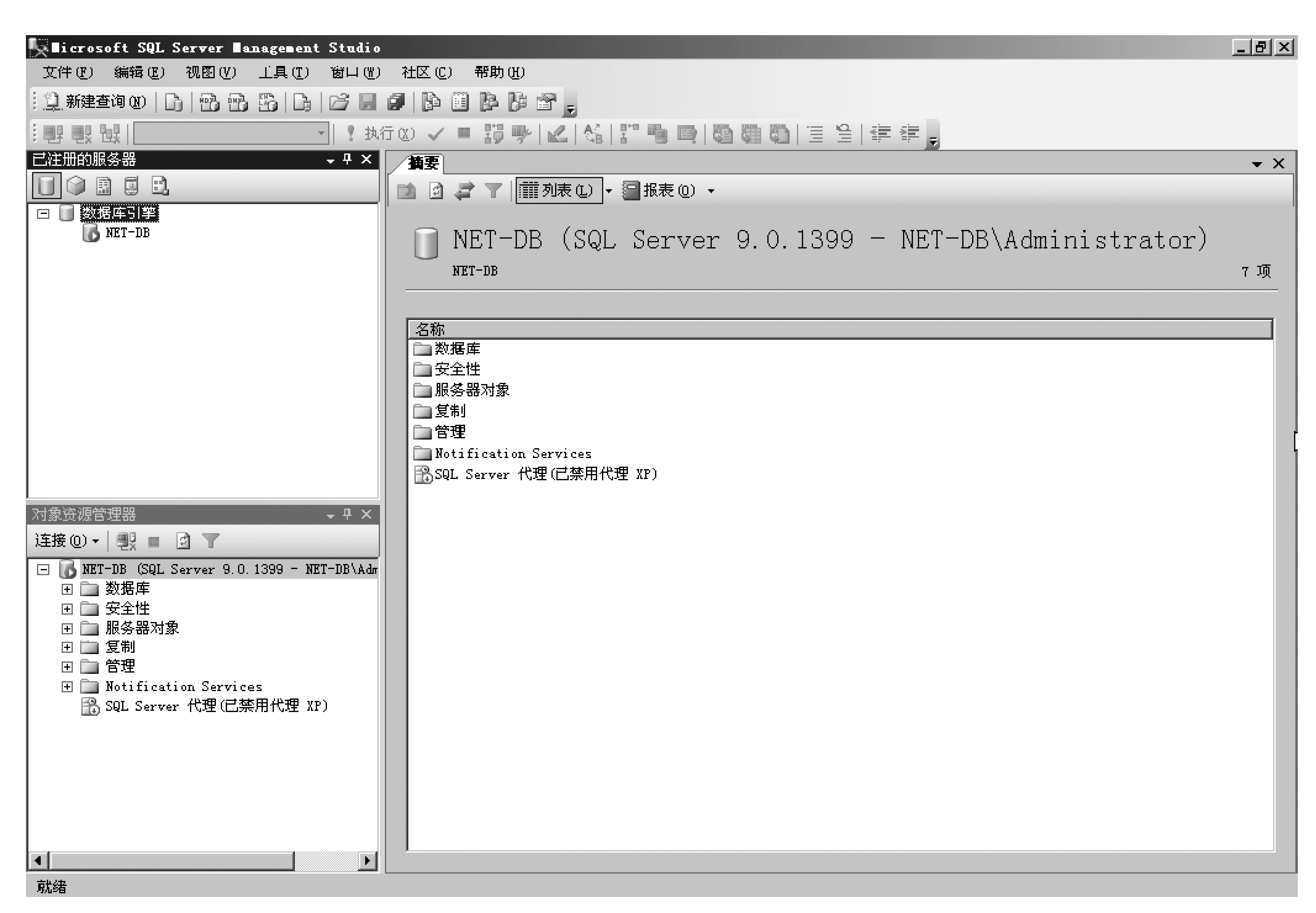


图 5-37　SQL Server Management Studio 的主界面

若需要某个组件窗口，可在【视图】菜单上单击相应组件的名称调出对应的组件窗口。若要显示查询编辑器窗口，单击工具栏上的 新建查询(N) 按钮，在打开的查询窗口中可以编



辑 T-SQL 命令。

5.3.4　经验总结

(1) 安装 Microsoft SQL Server 2005 时，若操作系统不同，可能会有不同的安装结果，读者需要注意的是如图 5-28 所示的【完成 Microsoft SQL Server 2005 安装】界面给出的安装提示信息，比如 Windows VISTA 操作系统，就会要求补丁，可以根据提示内容和给出的微软网站补丁网页的链接轻松地下载补丁，完成 Microsoft SQL Server 2005 的安装。提示信息会给的非常详细，按照提示信息的提示去操作，是很容易完成安装的。

(2) SQL Server 2005 服务器具有一定的自动配置能力，在安装后即使不做任何设置，它也能够正常运行，这样在大多数情况下，用户不必设置服务器选项。但是，如果系统默认值不适合自己的需要或为了更好地管理和优化 SQL Server 资源，读者可以使用 sp_configure 系统存储过程或 SQL Server Management Studio 对服务器配置选项进行设置。这里不做详述。

(3) 安装时指定的身份验证模式在以后使用 SQL Server 时，可以依照需要进行修改。这里的身份验证模式设置将影响“新闻类别网站”中的连接数据库代码的编写。

修改方法如下：在如图 5-35 所示的【对象资源管理器】中，右击 SQL Server 实例名（如 NET-DB）并在弹出菜单中选择【属性】命令，在打开的如图 5-38 所示的【服务器属性】|【选择页】面板中，选择【安全性】图标，并在【服务器身份验证】区域选择自己想要的身份验证模式。

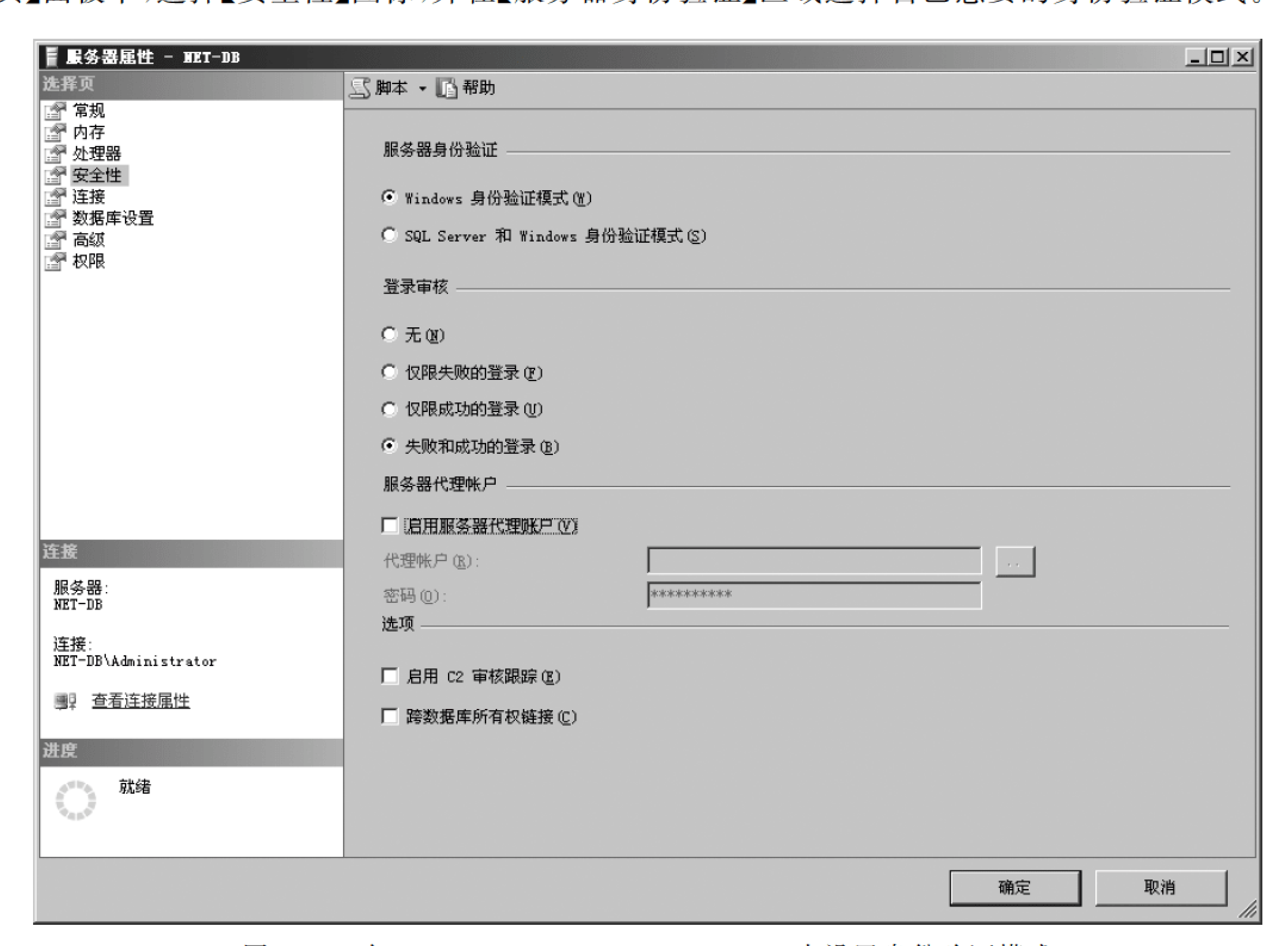
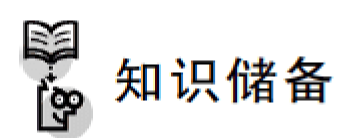


图 5-38　在 SQL Server Management Studio 中设置身份验证模式



5.4 安装、配置 Eclipse



知识储备

Eclipse 是开放源代码项目，可以免费下载，其官方网站网址是 http://www. eclipse. org。官方网站提供 Releases、Stable Builds、Integration Builds 和 Nightly Builds 等版本的下载。建议读者使用 Releases 或 Stable Builds 版本。Releases 版本是 Eclipses 开发团队发布的主要版本，是经过测试的稳定版本，适合要求稳定而不需要最新改进功能的使用者选择。目前最新的 Releases 版本是 Eclipse 3.4。

5.4.1 任务描述

构建“新闻类别网站”的开发环境之四：安装、配置 Eclipse。

5.4.2 任务实现

下面以在 Windows XP 系统中安装 Eclipse 为例，说明 Eclipse 的安装、配置过程。由于 Eclipse 的安装、配置过程比较复杂，因此分三个任务分别讲述。

任务 1　安装 Eclipse
任务 2　汉化 Eclipse
任务 3　配置 Eclipse

任务 1：安装 Eclipse

具体操作步骤如下：

(1) 登录 Eclipse 的官方站点 http://www. eclipse. org，下载 Eclipse 安装包 eclipse-SDK- Release 3. 4. 0-win32. zip。

(2) 将 ZIP 文件解压至自行设定的安装目录，如 D：\eclipse，Eclipse 即安装完毕，如图 5-39 所示。

任务 2：汉化 Eclipse

Eclipse 提供了一个语言包的插件，用于使其开发环境国际化。对于英文不好的读者或者初学者来说，中文版本的 Eclipse 可以显著提高学习效率，便于接受和使用 Eclipse。建议读者在对 Eclipse 有了初步的认识之后，还是多使用英文版 Eclipse，这样对以后的进一步学习和使用有好处。具体操作步骤如下：

(1) 在 Eclipse 的安装目录，如 D：\eclipse 中，找到 eclipse. exe，双击运行 Eclipse，启动后出现如图 5-40 所示的界面。

(2) 在【Workspace Launcher】界面中，提示“Select a workspace”，workspace（工作空间）是用来存储项目中的全部文件，在此保持默认设置。单击【OK】按钮。







图 5-39　解压后的 Eclipse 目录

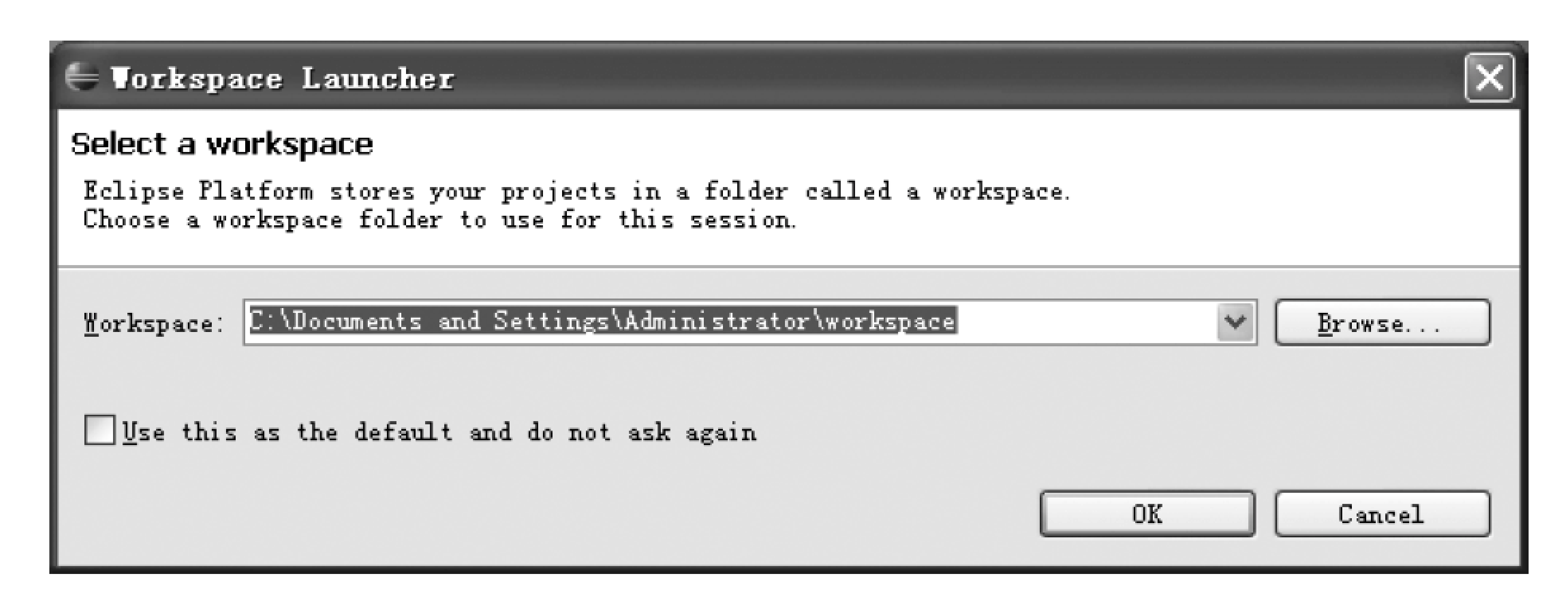


图 5-40　选择 Eclipse 工作空间

（3）Eclipse 启动后的界面如图 5-41 所示，单击 Eclipse 的菜单【Help】|【Software Updates】命令，弹出【Software Updates and Add-ons】对话框，如图 5-42 所示。

（4）单击选择【Available Software】，然后单击【Add Site】按钮，输入 URL：

```
http://download.eclipse.org/technology/babel/update-site/
```

或

```
http://download.eclipse.org/technology/babel/update-site/ganymede/
```

新输入的 URL 会加入到当前界面中，单击 URL 前面的"＋"号可以展开该网址对应的列表。若该网址对应的列表没有显示出来，单击【Refresh】按钮刷新一下。最后单击





图 5-41　Eclipse 启动后的界面

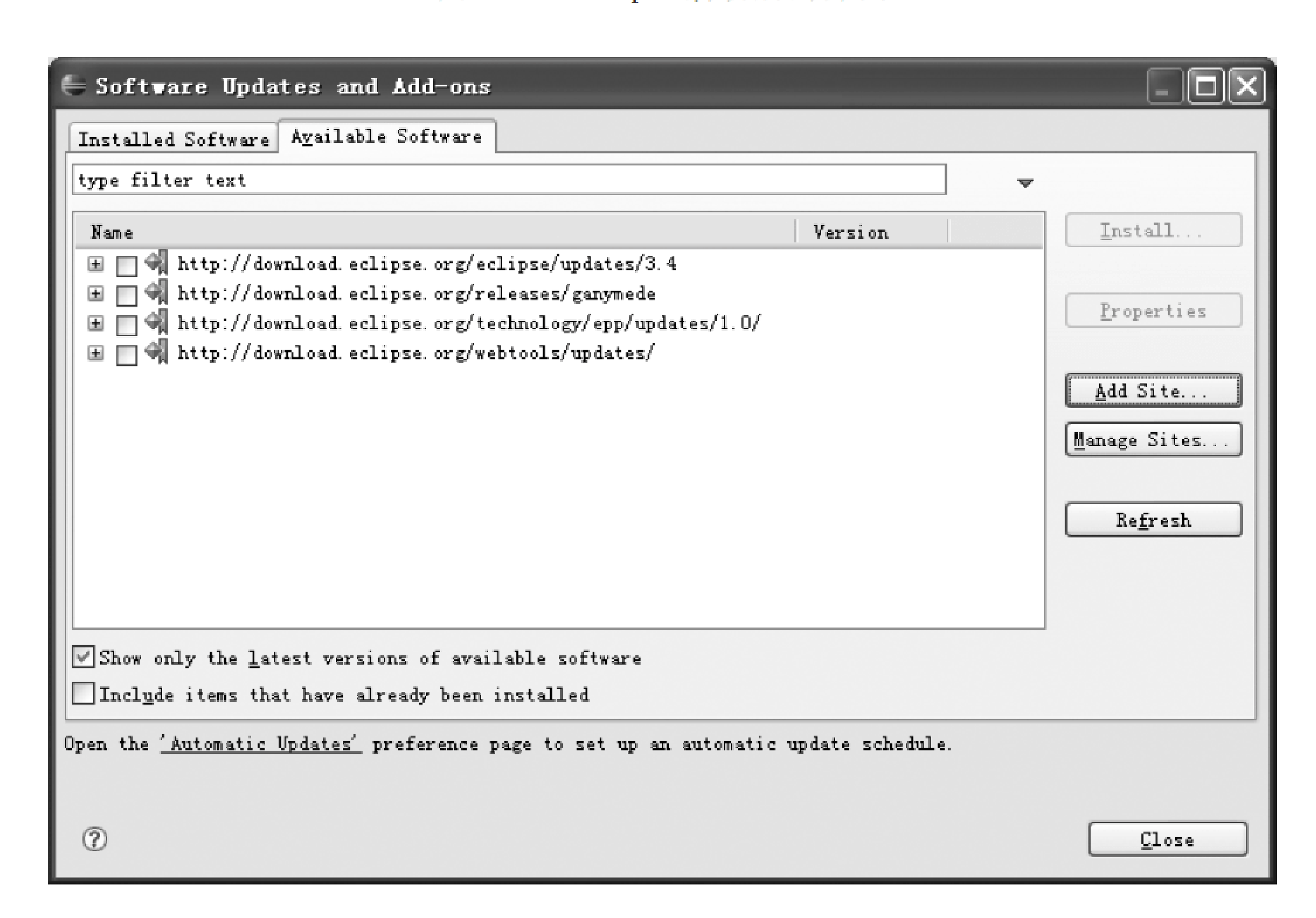


图 5-42　Eclipse 软件更新(Software Updates)界面

【Close】按钮。

(5) 在如图 5-43 所示的界面中选择【Eclipse Language Pack for Simplified Chinese】选项,单击【Next】按钮。

(6) 在如图 5-44 所示的【Eclipse 语言包安装协议】对话框中,选择【I accept the terms of the license agreement】单选按钮,接受许可协议,然后单击【Next】按钮。



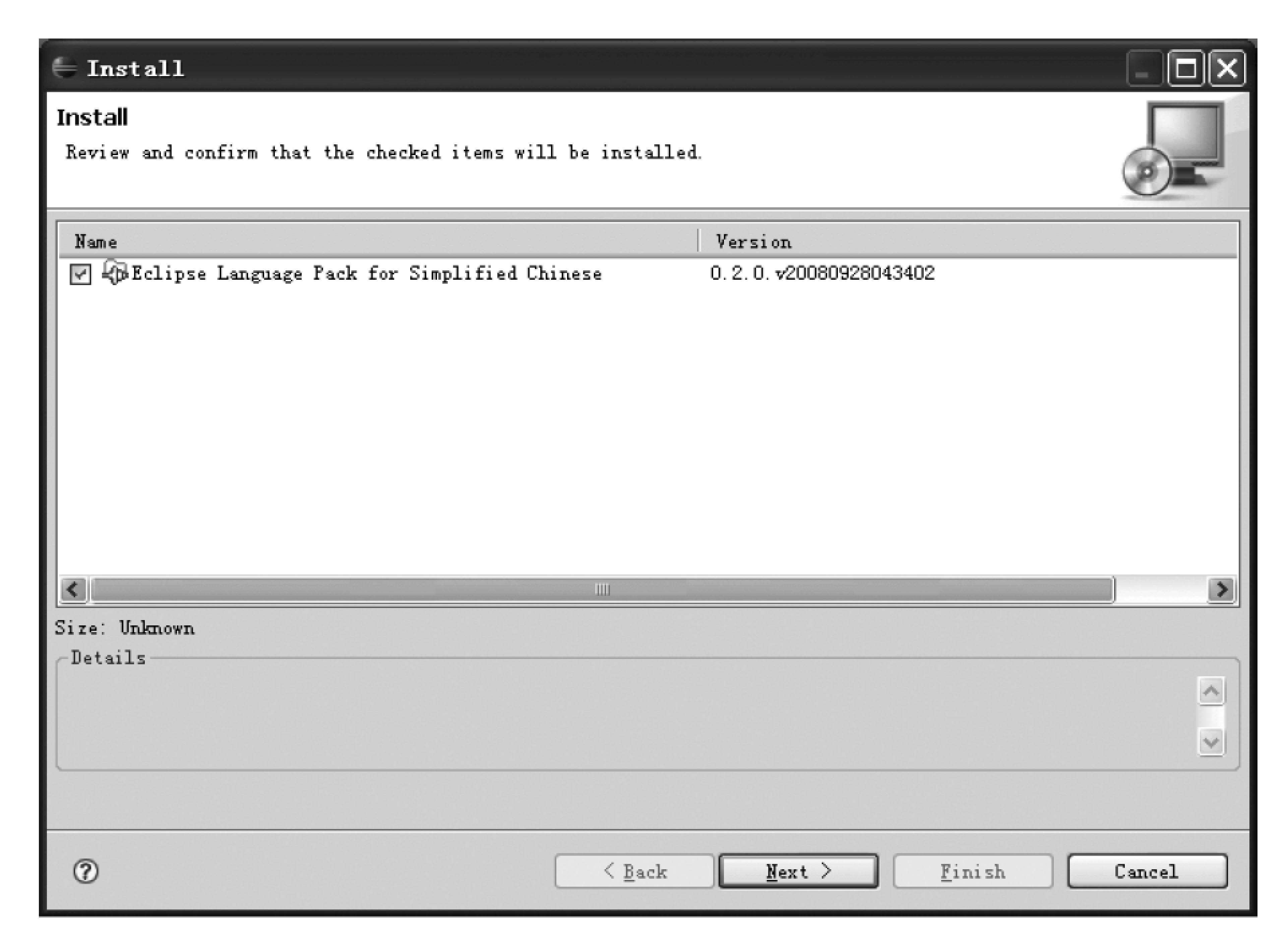


图 5-43　获取 Eclipse 语言包

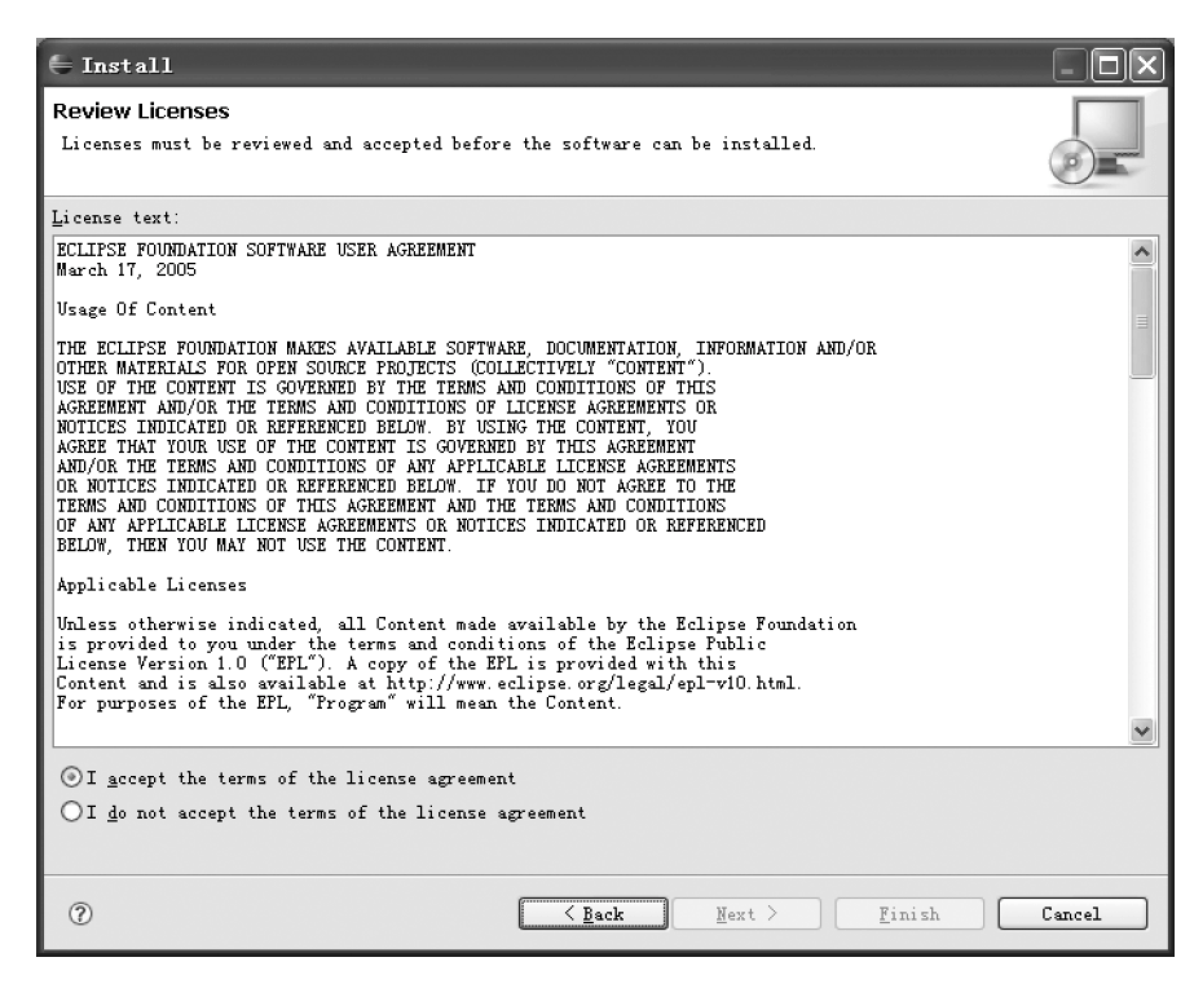


图 5-44　Eclipse 语言包安装协议

（7）此时将返回 Eclipse 软件更新(Software Updates)界面，要等待“漫长”的更新过程，如图 5-45 所示。



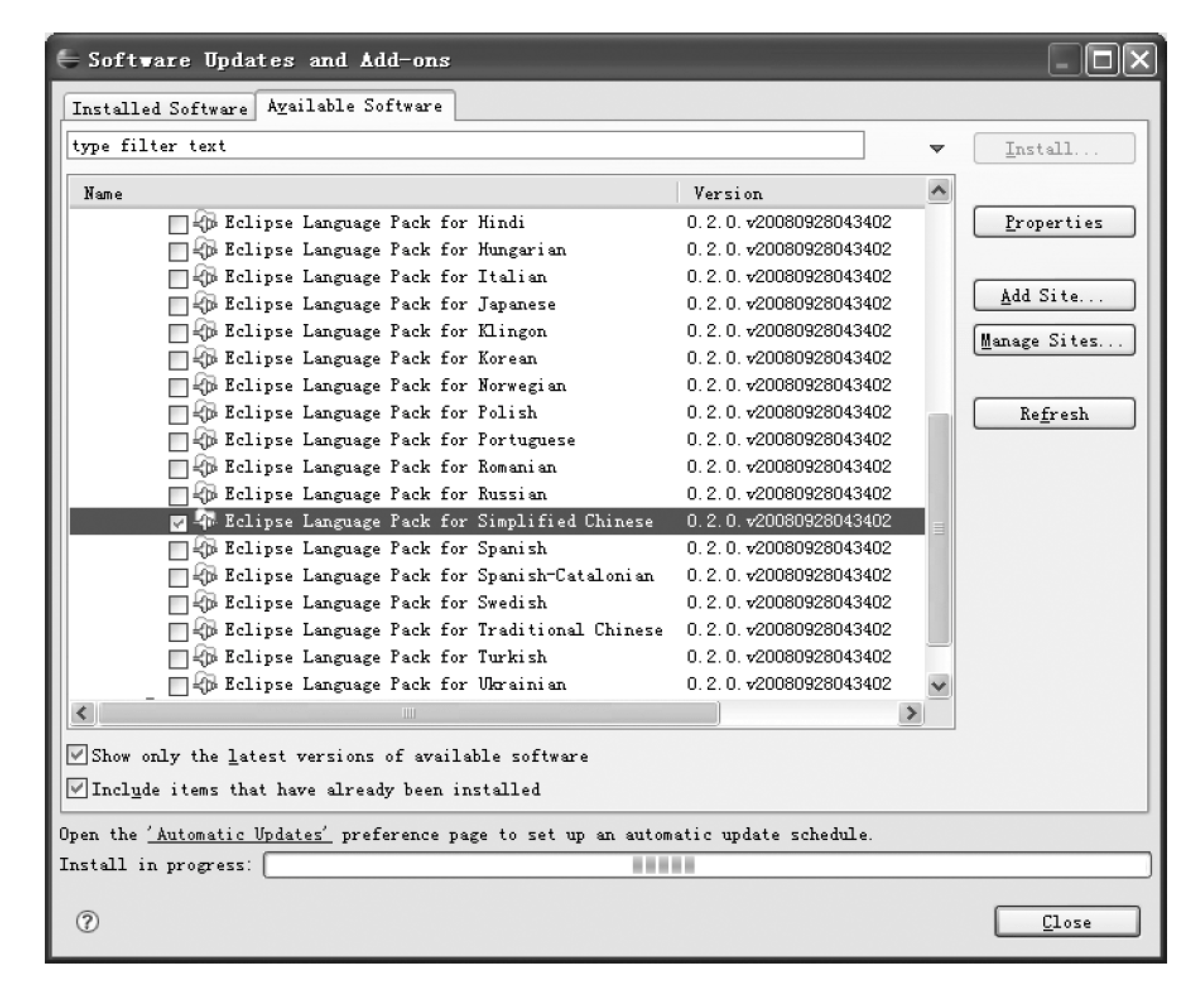


图 5-45　下载安装 Eclipse 语言包

(8) 安装完成后，重启 Eclipse，就是熟悉的中文界面了，如图 5-46 所示。和图 5-41 比较一下，可能初学者更喜欢这一图形界面。

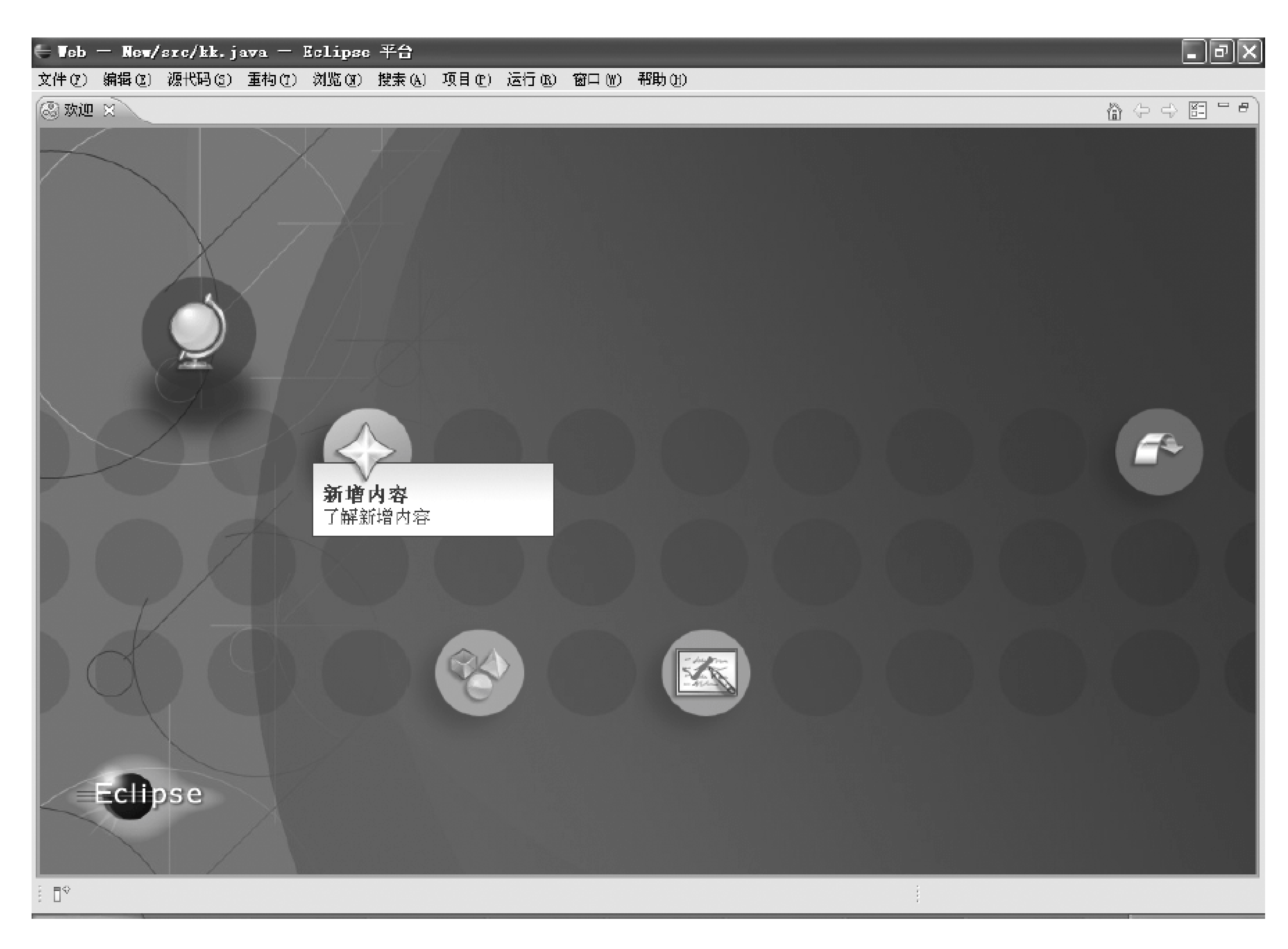


图 5-46　汉化后的 Eclipse



任务 3：配置 Eclipse

具体操作步骤如下：

（1）重启 Eclipse 后，选择【窗口】|【首选项】命令，出现如图 5-47 所示的对话框。

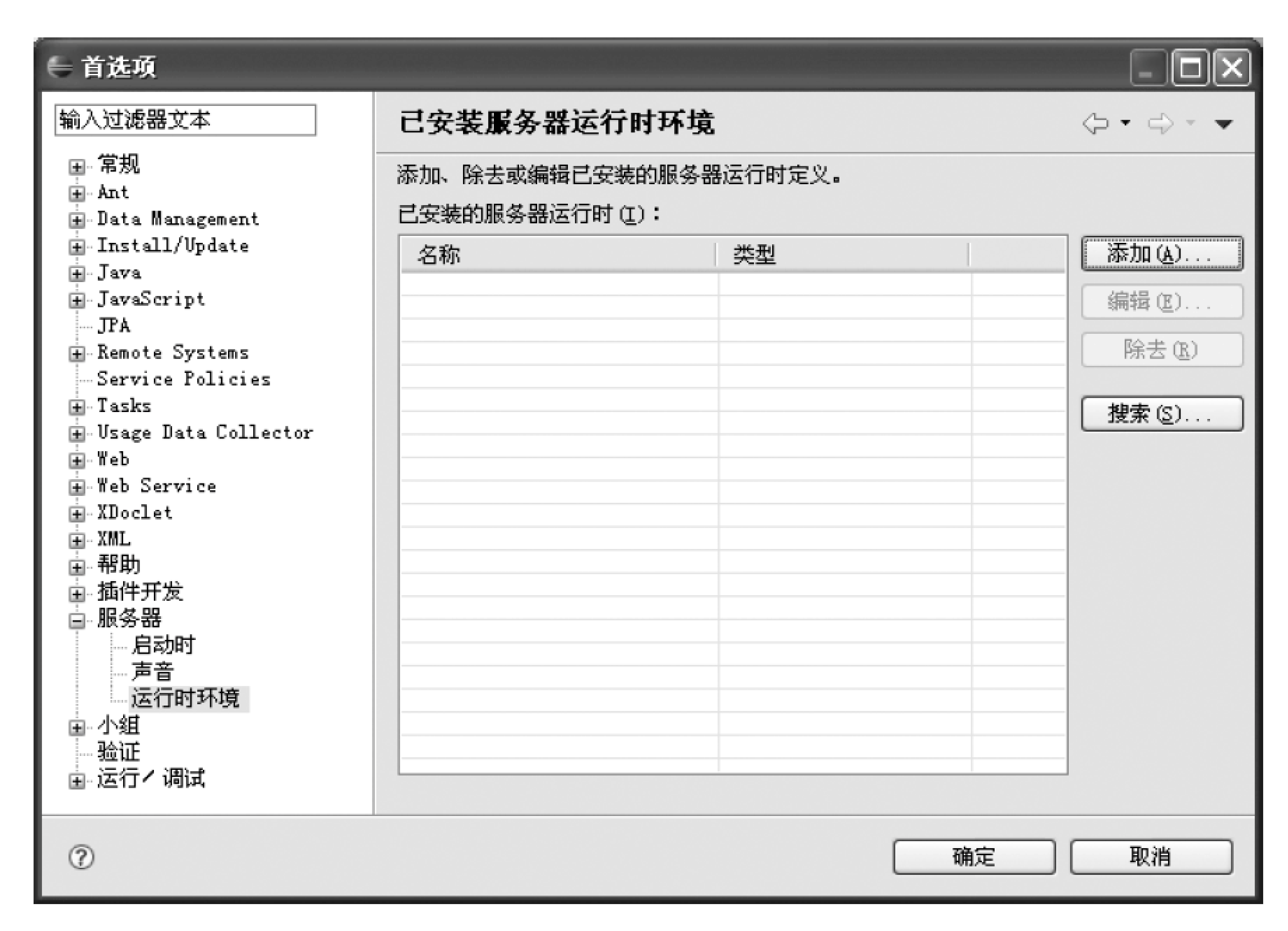


图 5-47　在【首选项】对话框中选择【运行时环境】选项

（2）在如图 5-47 所示的【首选项】对话框中，选择【服务器】|【运行时环境】命令。

（3）单击【添加】按钮，弹出【新建服务器运行时】对话框，如图 5-48 所示。单击【Apache Tomcat v6.0】选项后，再单击【下一步】按钮。



图 5-48　新建服务器运行时



（4）图 5-49 中，在【Tomcat 服务器指定安装目录】的【名称】文本框右侧单击【浏览】按钮，选择 Apache Tomcat v6.0 的安装目录。目录选择完成后，单击【确定】按钮，回到如图 5-50 所示的界面。



图 5-49　选择 Apache Tomcat v6.0 的安装目录

（5）在如图 5-50 所示的界面中，单击【完成】按钮，Eclipse 服务器的配置就完成了。配置完成的界面如图 5-51 所示。

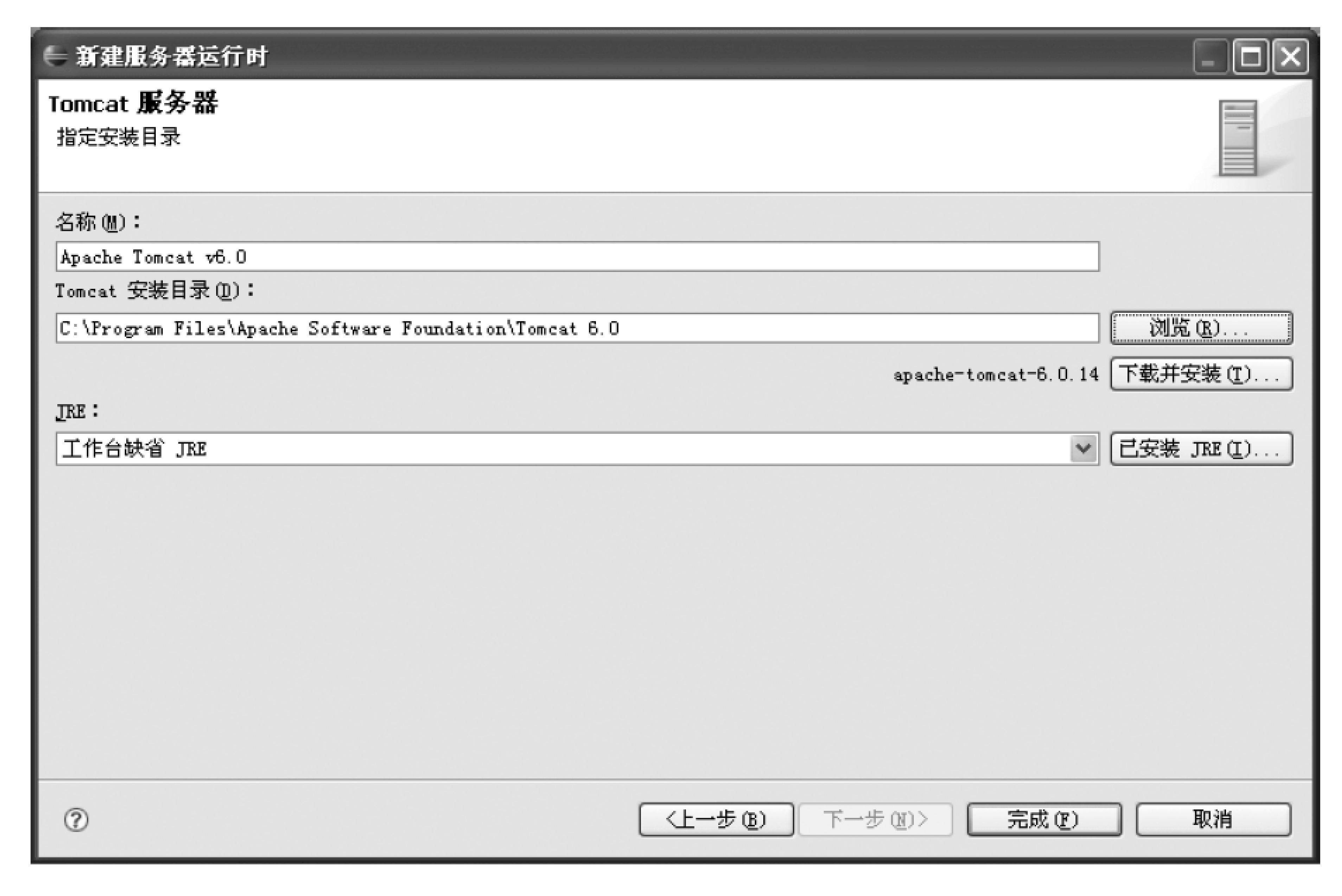


图 5-50　配置好的服务器运行时界面





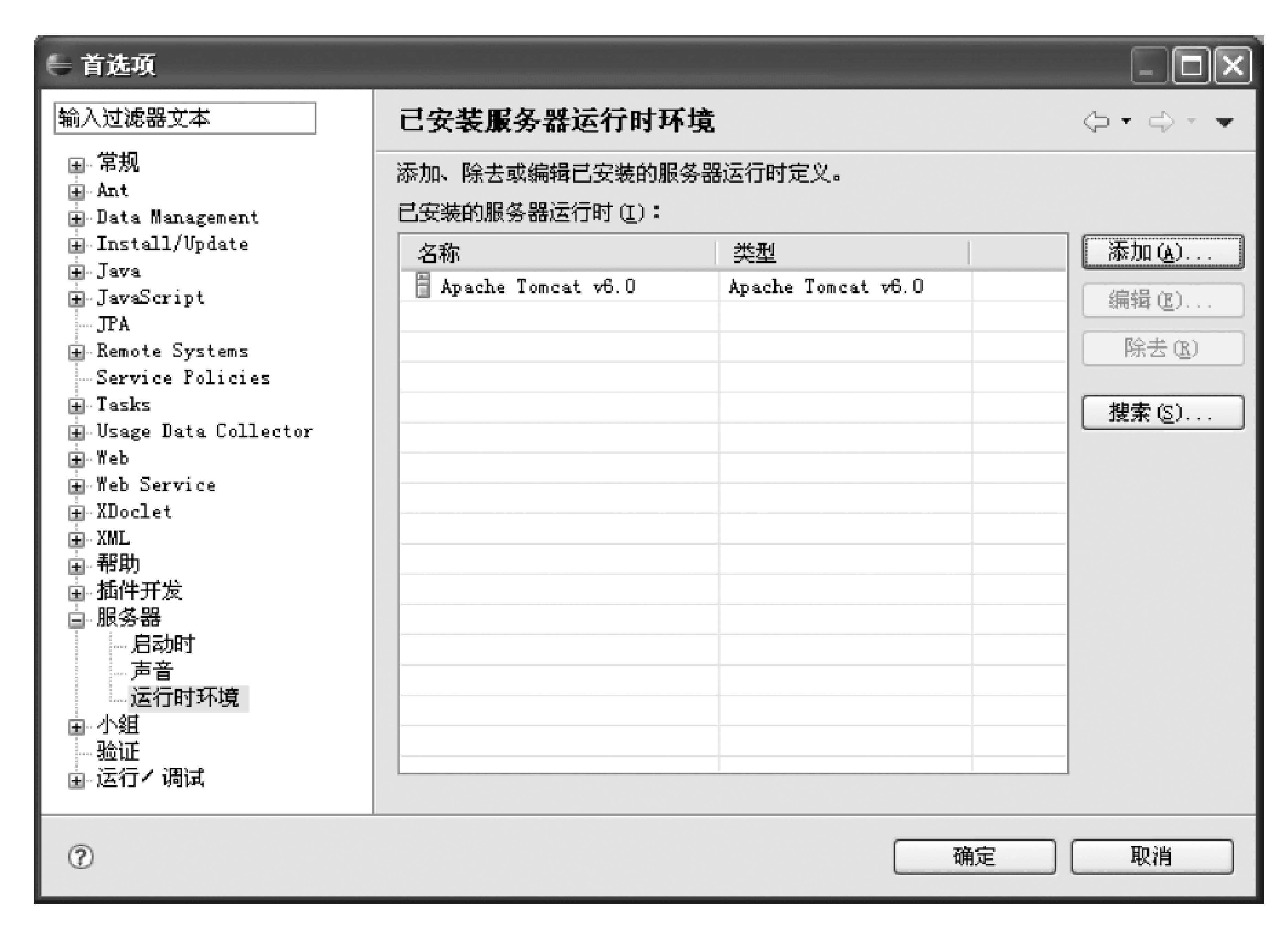


图 5-51　配置好的【运行时环境】对话框

5.4.3　相关知识拓展

Eclipse 最初是由 IBM 公司开发的替代商业软件 Visual Age for Java 的下一代 IDE 开发环境，2001 年 11 月贡献给开源社区，现在它由非赢利软件供应商联盟 Eclipse 基金会(Eclipse Foundation)管理。

Eclipse 是著名的跨平台的自由 IDE(Integrated Development Environment，集成开发环境)，最初主要用于 Java 语言开发，但是目前也有人通过插件使其作为其他计算机语言，比如 C++ 和 Python 的开发工具。Eclipse 的本身只是一个框架平台，但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。许多软件开发商以 Eclipse 为框架开发自己的 IDE。

虽然大多数用户很乐于将 Eclipse 当做 Java IDE 来使用，但 Eclipse 的目标不仅限于此。Eclipse 还包括插件开发环境(Plug-in Development Environment，PDE)，这个组件主要针对希望扩展 Eclipse 的软件开发人员，允许他们构建与 Eclipse 环境无缝集成的工具。

Eclipse 的设计思想是"一切皆为插件"。Eclipse 核心非常小，其他功能都基于此核心做成插件，比如 Eclipse 的图形 API(称为 SWT/JFace)、Java 开发环境插件(简称 JDT)、插件开发环境(简称 PDE)等。Eclipse 还对这些插件的协同工作提供了良好的支持，不仅安装简单，还可以无缝结合。Eclipse 的体系结构如图 5-52 所示。

Eclipse 对这些插件是动态载入并动态调用的。Eclipse 启动后只有在真正用到某个插件时，该插件才会被调入内存，当该插件不再被使用时，它就会在适当的时候被清除出内存。因此即使装了一大堆插件在 Eclipse 中，也不必担心某些不常用的插件白白消耗内存。

基于 Eclipse 的应用程序的突出例子是 IBM 的 WebSphere Studio Workbench，它构成



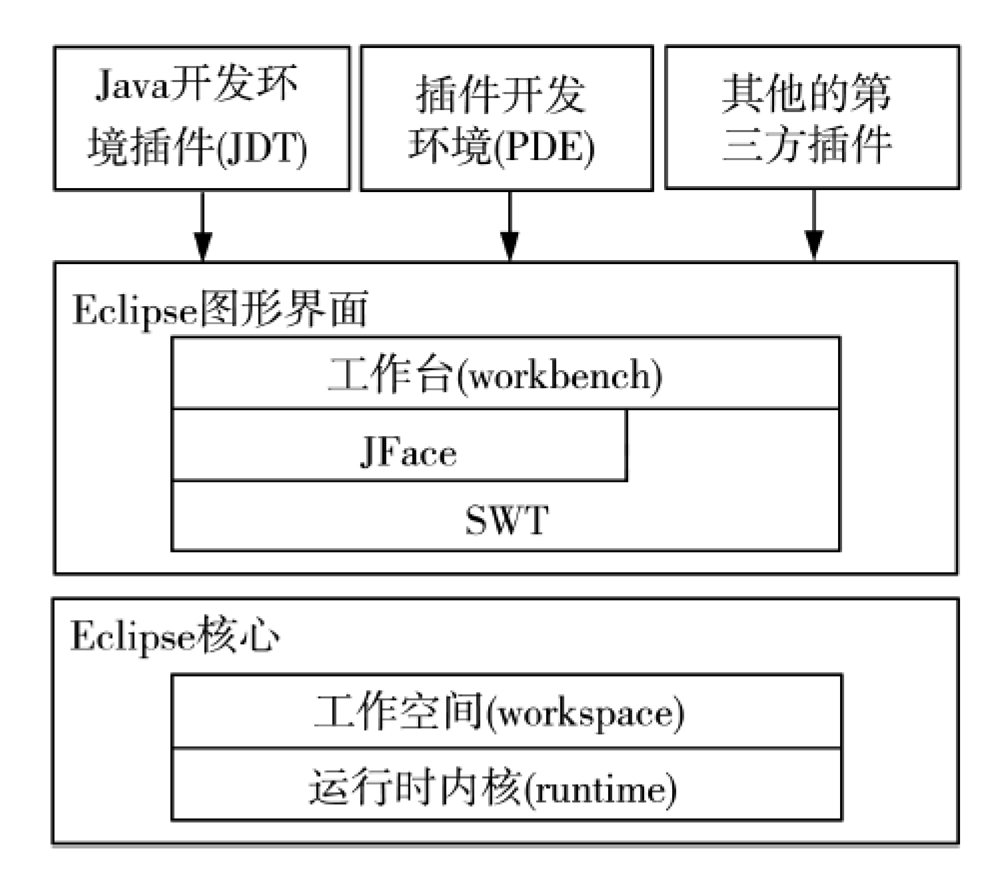


图 5-52　Eclipse 的体系结构

了 IBM Java 开发工具系列的基础。例如，WebSphere Studio Application Developer 添加了对 JSP、Servlet、EJB、XML、Web 服务和数据库访问的支持。

5.4.4　经验总结

Eclipse 是用 Java 编写的，所以它的运行需要 Java 运行环境（Java Runtime Environment）的支持，故在安装 Eclipse 前必须先安装 JDK；对于 J2EE 的开发必须提供运行 J2EE 的 Web 应用服务器，故需提前安装 Web 应用服务器，如 Tomcat 服务器。

5.5　重新配置 Dreamweaver 站点

5.5.1　任务描述

构建“新闻类别网站”的开发环境之五：配置 Dreamweaver。

5.5.2　任务实现

具体操作步骤如下：

（1）重新打开 Dreamweaver，在菜单栏中依次选择【站点】|【管理站点】命令，打开【管理站点】对话框，如图 5-53 所示。

（2）选中在第 4 章建立的【JSP 新闻系统】站点，单击对话框右侧的【编辑】按钮。

（3）弹出如图 5-54 所示的【JSP 新闻系统的站点定义为】对话框，在【您的站点的 HTTP 地址（URL）是什么?】文本框输入“http://localhost：8080/New/”，单击【下一步】按钮。



图 5-53　【管理站点】对话框







图 5-54　设置站点的 HTTP 地址

（4）出现如图 5-55 所示的对话框，在此设置站点使用的服务器技术，由于我们要使用



图 5-55　设置站点的服务器技术



服务器端的 JSP 技术来进行"新闻类别网站"的开发，所以选择【是，我想使用服务器技术】并在【哪种服务器技术?】中选择【JSP】选项，单击【下一步】按钮。

(5) 出现设置站点文件存储位置的对话框，选择【在本地进行编辑和测试(我的测试服务器是这台计算机)】单选按钮(根据实际需要，也可以选择另外两个单选按钮进行相应的设置)，并在【您将把文件存储在计算机上的什么位置?】文本框中输入 Tomcat 服务器的网站根目录或某个虚拟目录的路径，或者通过单击文件夹图标来选择路径，如图 5-56所示。



图 5-56 设置站点文件存储位置

(6) 单击【下一步】按钮，出现如图 5-57 所示的测试服务器 URL 对话框，输入站点根文件夹的 URL，并单击【测试 URL】按钮，当看到"URL 前缀测试已成功"信息时，单击【确定】按钮以关闭弹出的对话框。

(7) 单击【下一步】按钮，出现如图 5-58 所示的远程服务器设置对话框，根据实际情况选择下列选项之一：

- 如果编辑完一个文件后，要把该文件复制到另一台计算机，可选择【是的，我要使用远程服务器】单选按钮。
- 如果不使用远程服务器，可选择【否】单选按钮。

此处，我们选择【否】单选按钮完成设置，然后单击【下一步】按钮。

(8) 如图 5-59 所示，Dreamweaver 列出站点的设置信息，核查后单击【完成】按钮。

(9) Dreamweaver 将重新建立站点的缓存区，使上述设置生效，如图 5-60 所示。







图 5-57　测试服务器 URL



图 5-58　远程服务器设置





图 5-59　站点的设置信息



图 5-60　重新建立站点的缓存区



5.5.3　相关知识拓展

Web 站点是一组具有共享属性(如相关主题、类似的设计或共同目的)的链接文档和资源。Dreamweaver 是创建和管理站点的工具,使用它不仅可以创建单独的文档,还可以创建完整的 Web 站点。

在 Dreamweaver 中,术语"站点"指属于某个 Web 站点的文档的本地或远程存储位置。Dreamweaver 站点提供了一种方法,利用它可以组织和管理站点所有的 Web 文档、将站点上传到 Web 服务器、跟踪和维护网页链接、管理和共享文件。

注意

若要定义 Dreamweaver 站点,只需设置一个本地文件夹。但若要向 Web 服务器传输文件或开发 Web 应用程序,还必须添加远程站点和测试服务器信息。

5.5.4　经验总结

1. 应用程序服务器

应用程序服务器是一种软件,用来帮助 Web 服务器处理动态页。选择应用程序服务器时,应该考虑多种因素,包括预算、要使用的服务器技术(ColdFusion、ASP. NET、ASP、JSP 或 PHP)以及 Web 服务器的类型。

应用程序服务器使用不同的服务器技术。Dreamweaver 支持五种服务器技术:ColdFusion、ASP. NET、ASP、JSP 和 PHP。表 5-2 列出了可用于 Dreamweaver 支持的五种服务器技术的常见应用程序服务器。

表 5-2　支持不同服务器技术的应用程序服务器

服务器技术	服　务　器
ColdFusion	Macromedia ColdFusion MX 7
ASP. NET	具有 .NET 框架的 Microsoft IIS 6
ASP	Microsoft IIS Sun ONE Active Server Page
JSP	Macromedia JRun Apache Tomcat
PHP	PHP 服务器

注意

确保应用程序服务器可以和 Web 服务器配合使用。例如,.NET 框架只能和 IIS 5 或更高版本一起使用。

2. Dreamweaver 站点中设置连接远程服务器的方式

Dreamweaver 站点中设置连接远程服务器的方式有:无、FTP、本地/网络、WebDAV、



RDS、SourceSafe(R)数据库六种，可根据具体的情况选择，不同选择的设置也不同，要仔细揣摩。

5.6 测试搭建的开发环境

5.6.1 任务描述

构建“新闻类别网站”的开发环境之六：测试搭建的开发环境。

5.6.2 任务实现

按照上面的步骤搭建了开发环境后，要对搭建好的环境进行全面的测试，以确保开发环境的正常运行。这一步是很重要的，由于在搭建的过程中可能会有一些设置没有考虑到或者设置不正确，导致留下潜在的隐患，这些隐患可能会在我们实际开发的时候爆发。具体的环境测试分为三个任务来完成。

任务1　新建 Web 项目并进行设置
任务2　新建 JSP 文件并运行
任务3　新建 Servlet 文件并运行

任务1：新建 Web 项目并进行设置

具体操作步骤如下：

(1) 在汉化了的 Eclipse 软件中将“欢迎界面”关掉后，出现 J2EE 开发界面，如图 5-61 所示。

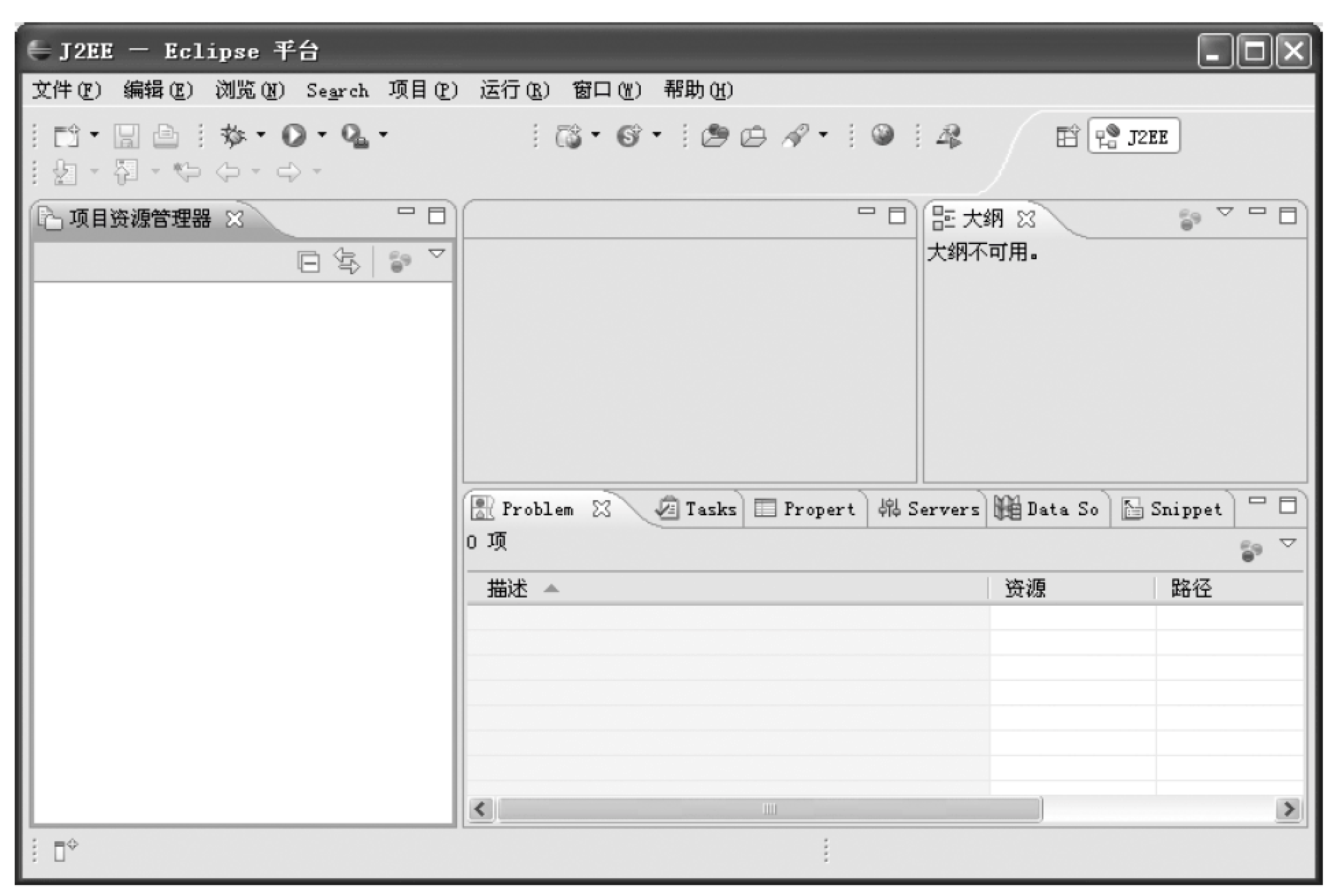


图 5-61　J2EE 开发界面

(2) 依次选择菜单【文件】|【新建】|【Dynamic Web Project】命令，打开 New Dynamic Web Project 对话框，如图 5-62 所示。

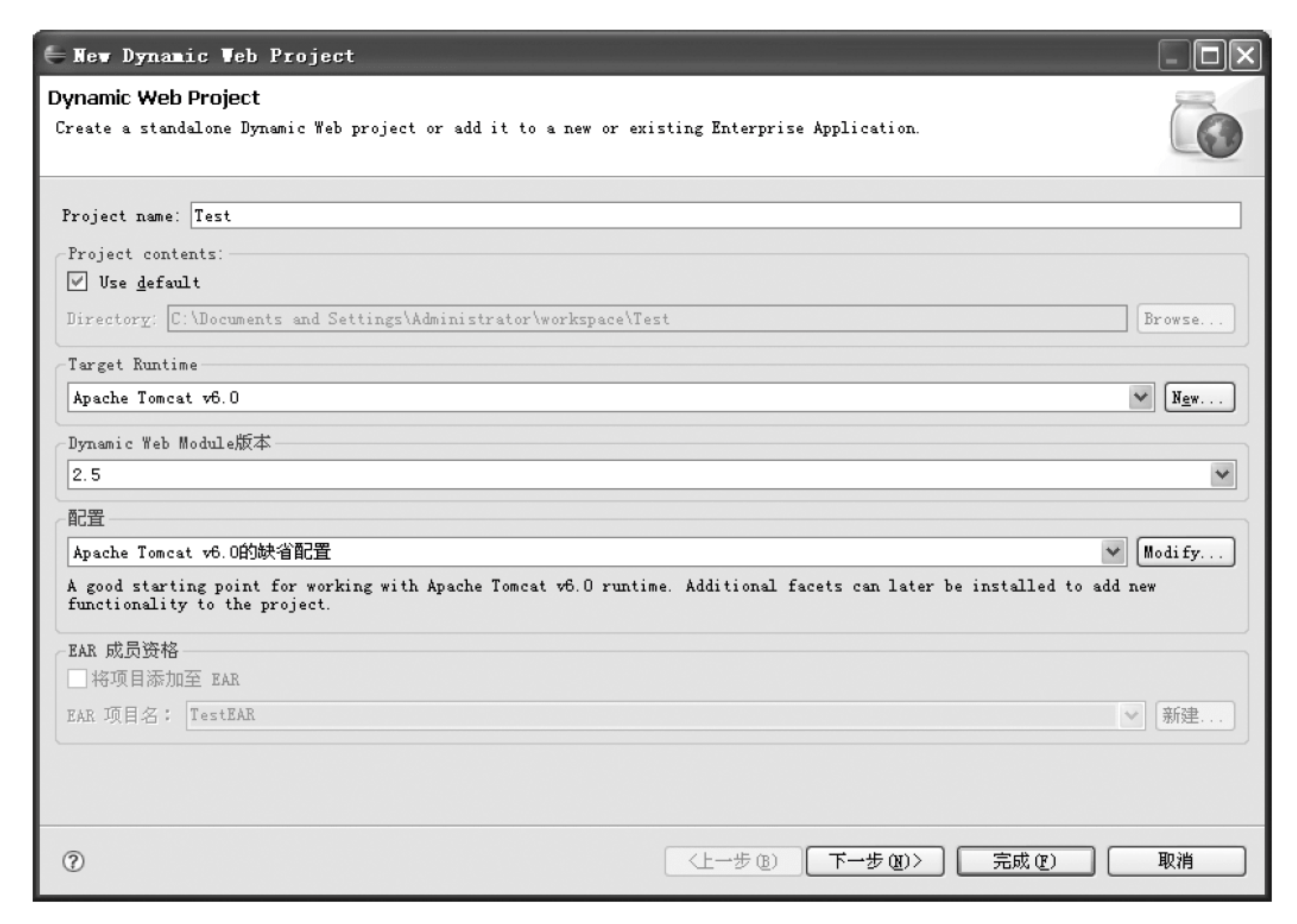


图 5-62　【New Dynamic Web Project】对话框

(3) 如图 5-62 所示，在弹出的【New Dynamic Web Project】对话框中，单击【Project name】文本框，输入 Web 项目名称“Test”；从【Target Runtime】列表框中选择已有的服务器，如 Apache Tomcat v6.0，或者单击【New】按钮来新建服务器。

(4) 然后一直单击【下一步】按钮，直到最后单击【完成】按钮，就完成了一个 Web 项目的创建，如图 5-63 所示。

任务 2：新建 JSP 文件并运行

具体操作步骤如下：

(1) 在 Eclipse 界面左侧的【项目资源管理器】上“右击”并选择新建的 Web 项目“Test”，再选择快捷菜单中的【新建】|【JSP】命令。

(2) 如图 5-64 所示，在弹出的【新建 Java Server 页】对话框中，在【文件名】文本框中输入“Default”，单击【下一步】按钮，直到单击【完成】按钮。

(3) Eclipse 自动打开刚刚新建的 Default.jsp 文件，如图 5-65 所示，在右侧的工作区中输入相应的 JSP 代码并保存。

(4) 在 Eclipse 界面左侧的“项目资源管理器”中右击鼠标并选择新建的 Default.jsp 文件，在快捷菜单中再选择【运行方式】|【在服务器上运行】命令，如图 5-66 所示。



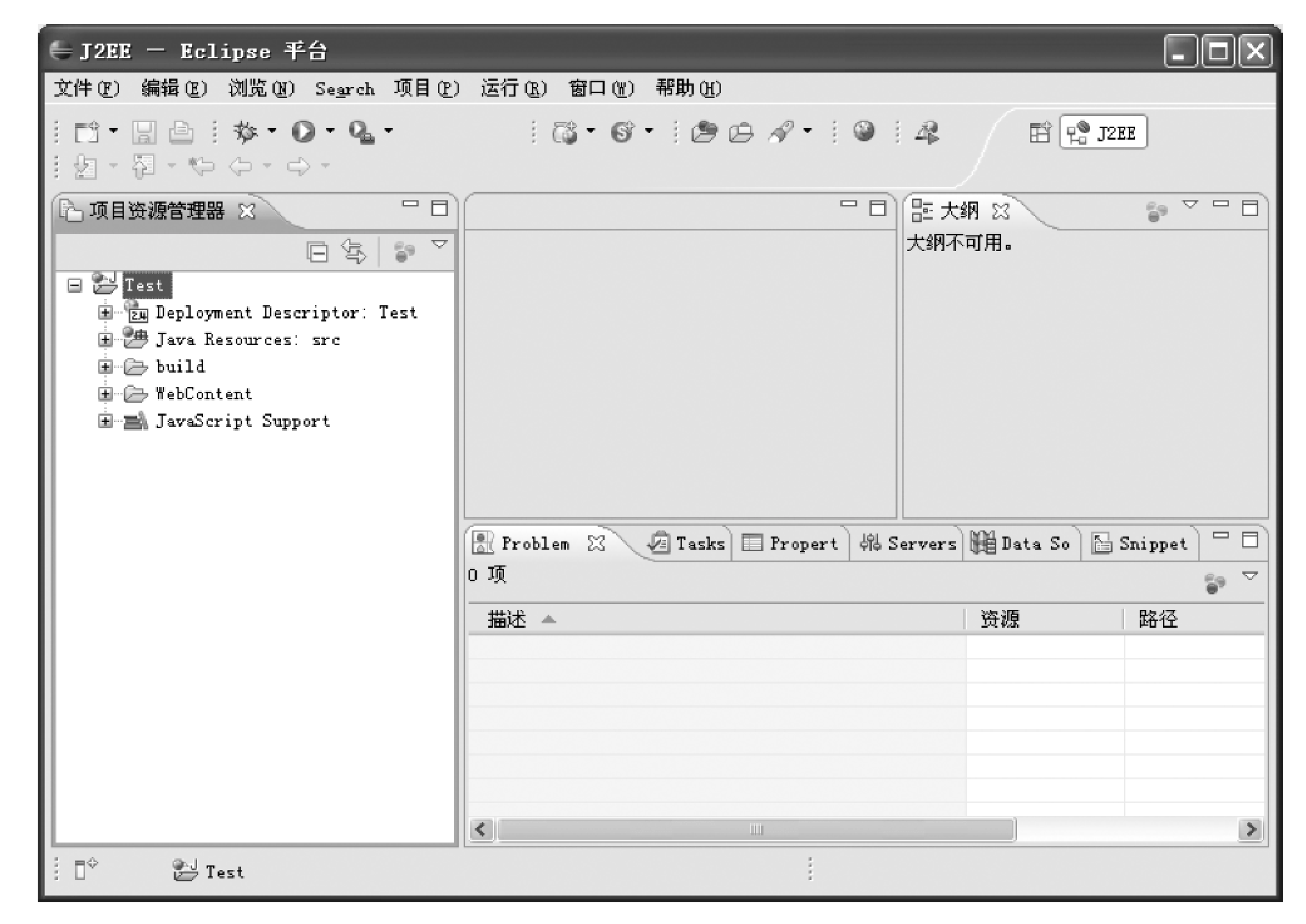


图 5-63　项目建成后的 Eclipse 界面

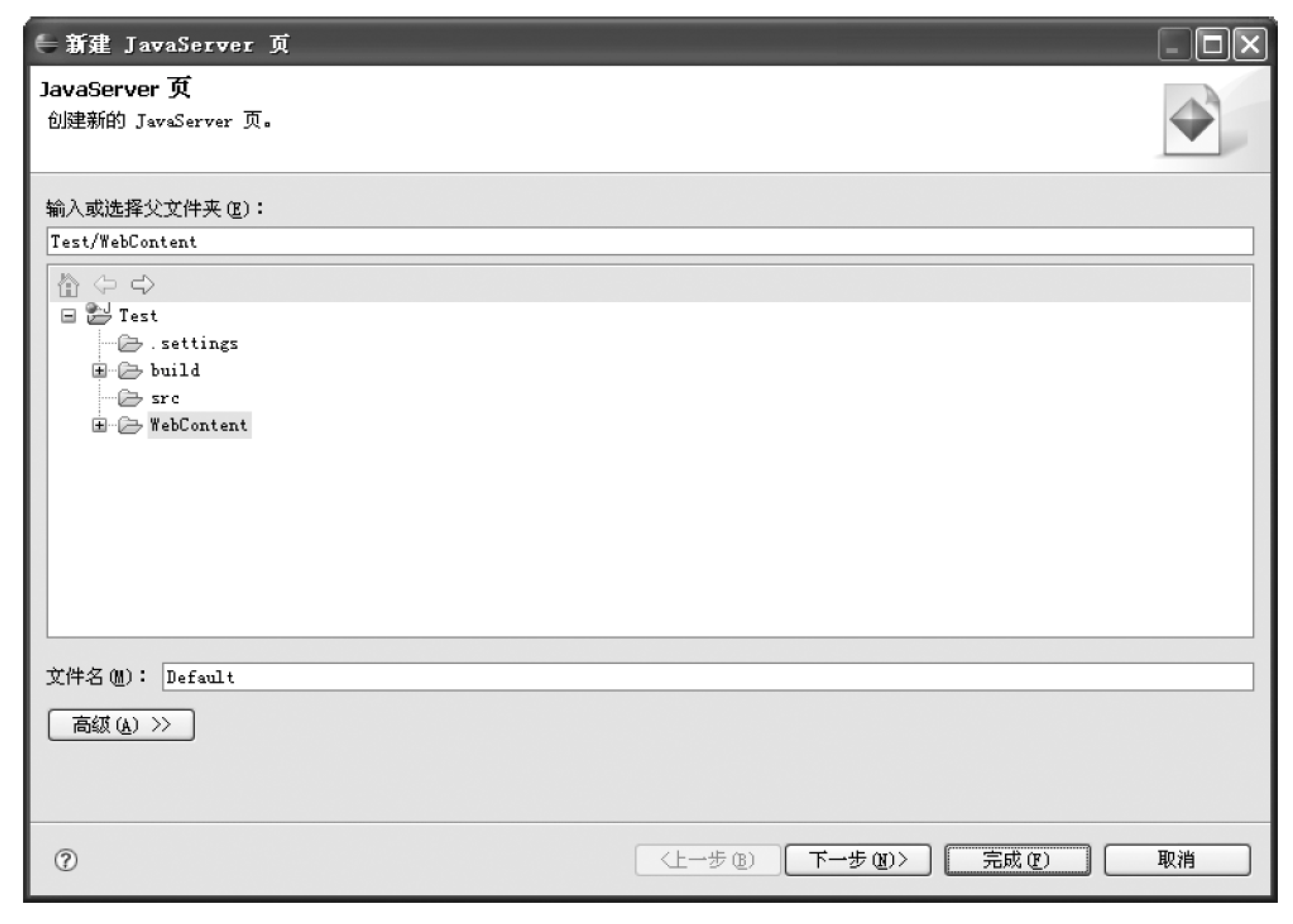


图 5-64　【新建 Java Server 页】对话框



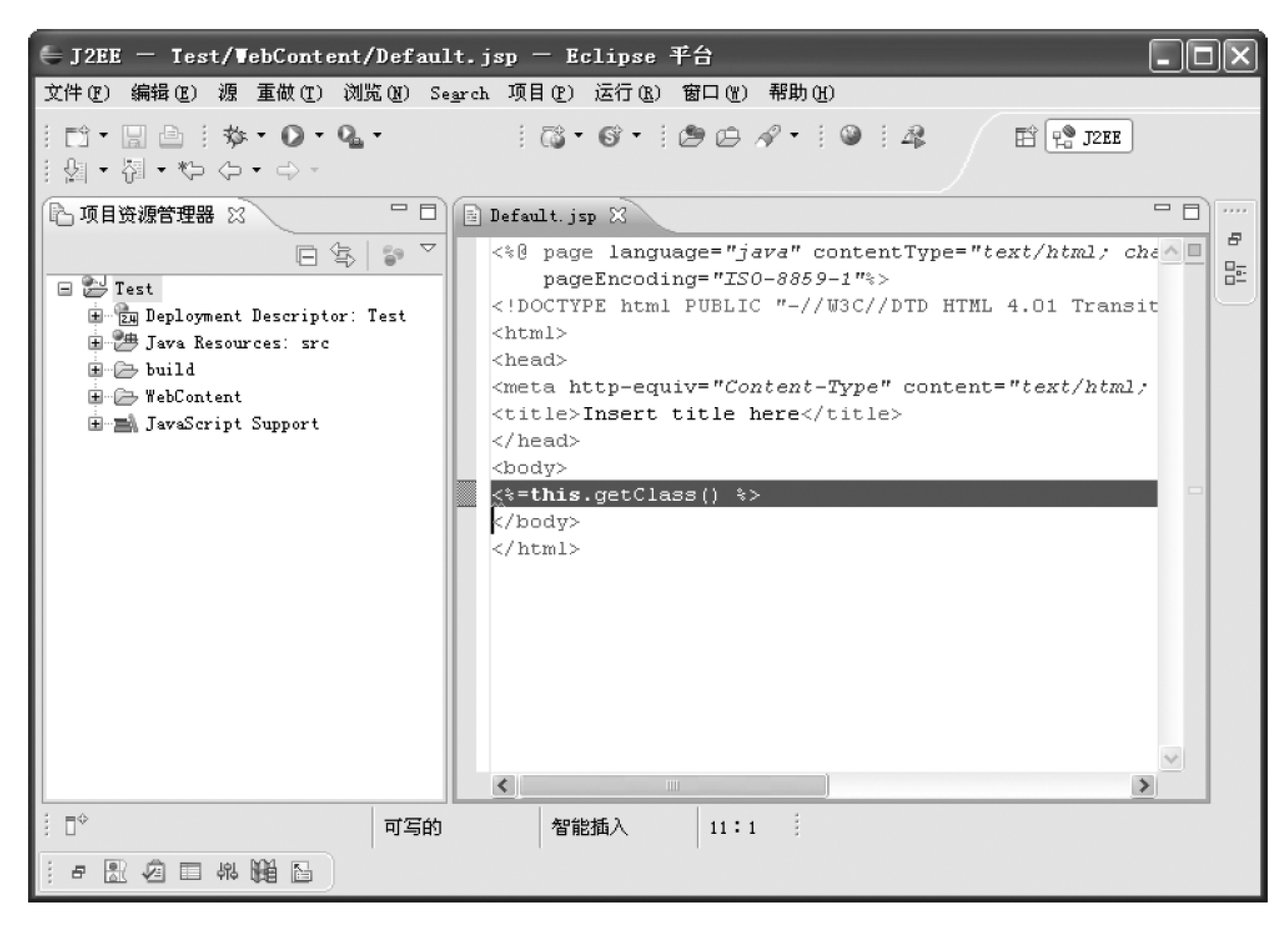


图 5-65　编辑 Default. jsp

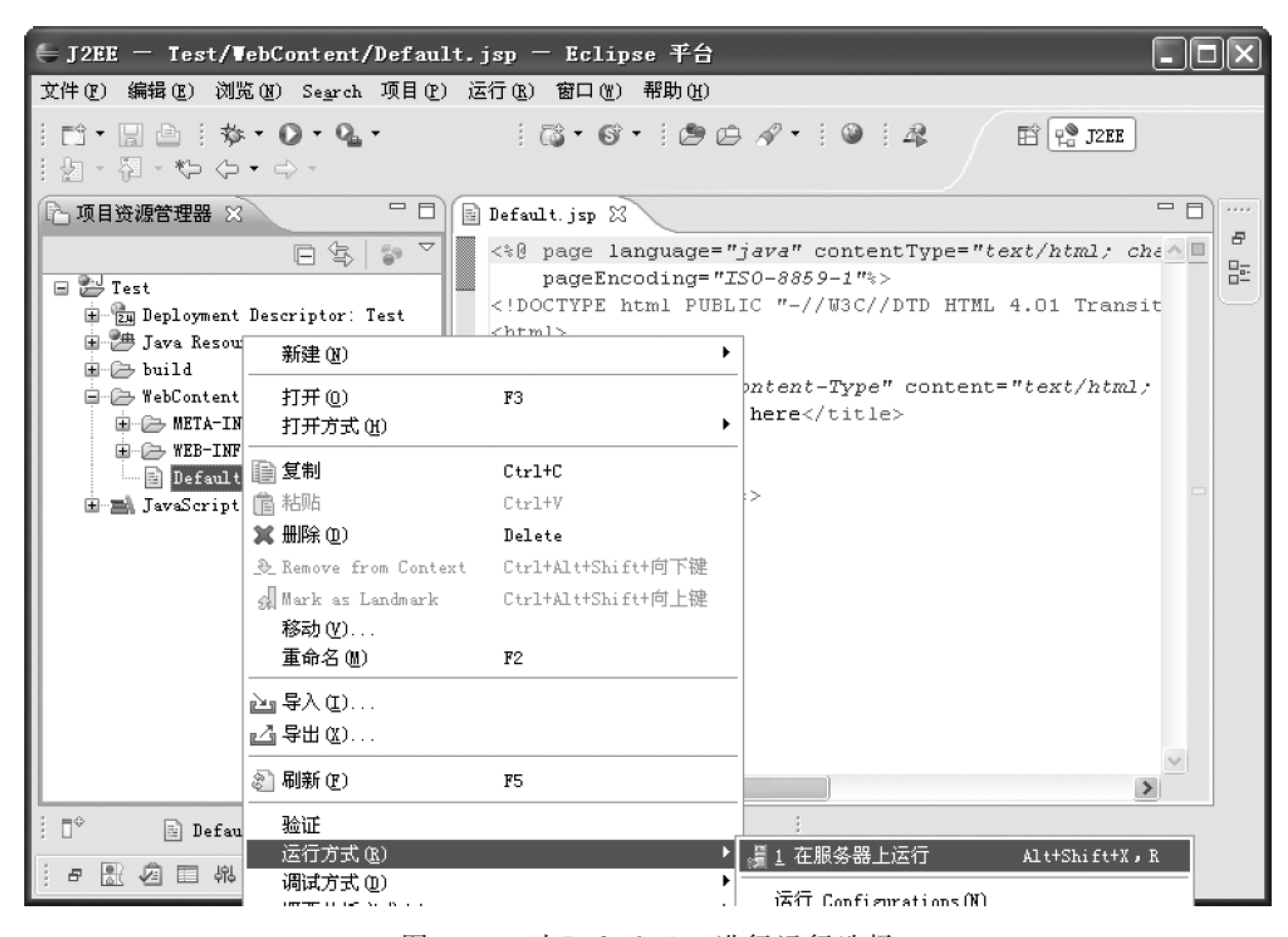


图 5-66　对 Default. jsp 进行运行选择



(5) 为当前运行的 JSP 页面配置服务器运行时，如果没有，可以单击【添加】新建服务器运行时，如图 5-67 所示，最后单击【完成】按钮。

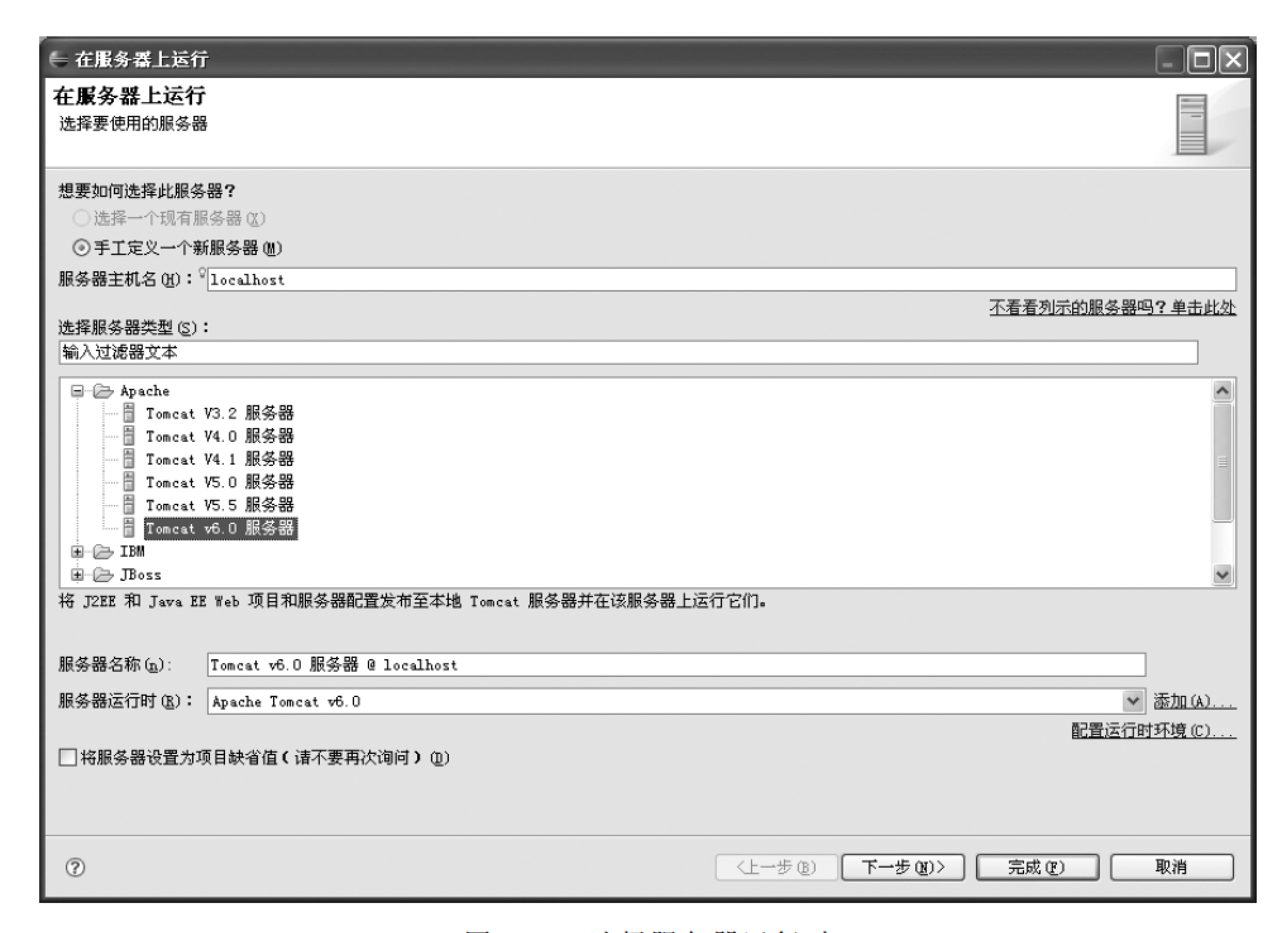


图 5-67　选择服务器运行时

(6) Eclipse 运行服务器，并显示运行结果，如图 5-68 所示。

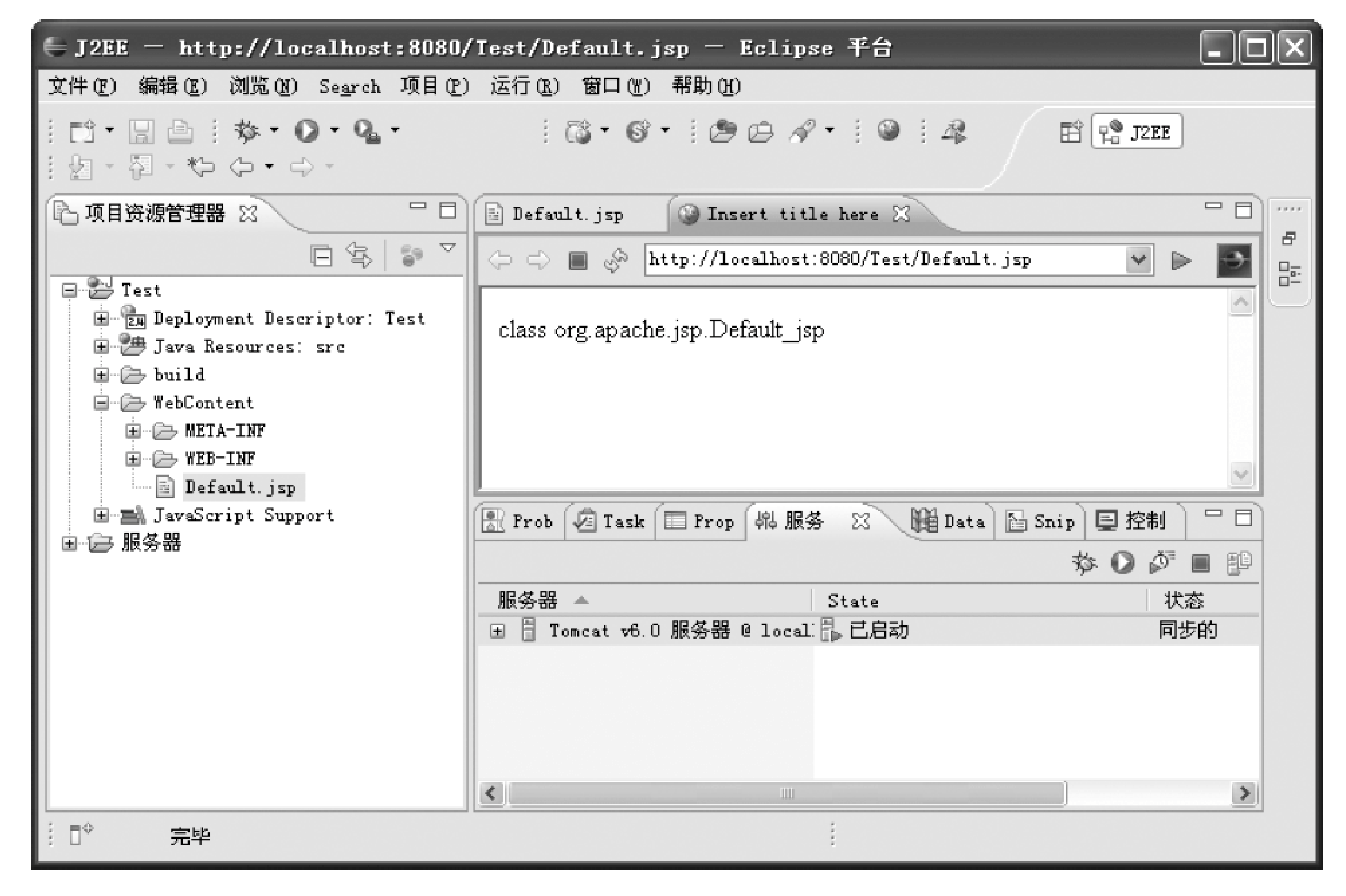


图 5-68　Default.jsp 运行后的结果





任务 3：新建 Servlet 文件并运行

具体操作步骤如下：

（1）在 Eclipse 界面左侧的【项目资源管理器】中右击并选择新建的 Web 项目“Test”，在快捷菜单中再选择【新建】|【Servlet】命令。

（2）如图 5-69 所示，在弹出的【Create Servlet】对话框中，在【Class name】文本框中输入“TestServlet”，单击【下一步】按钮，直到单击【完成】按钮。

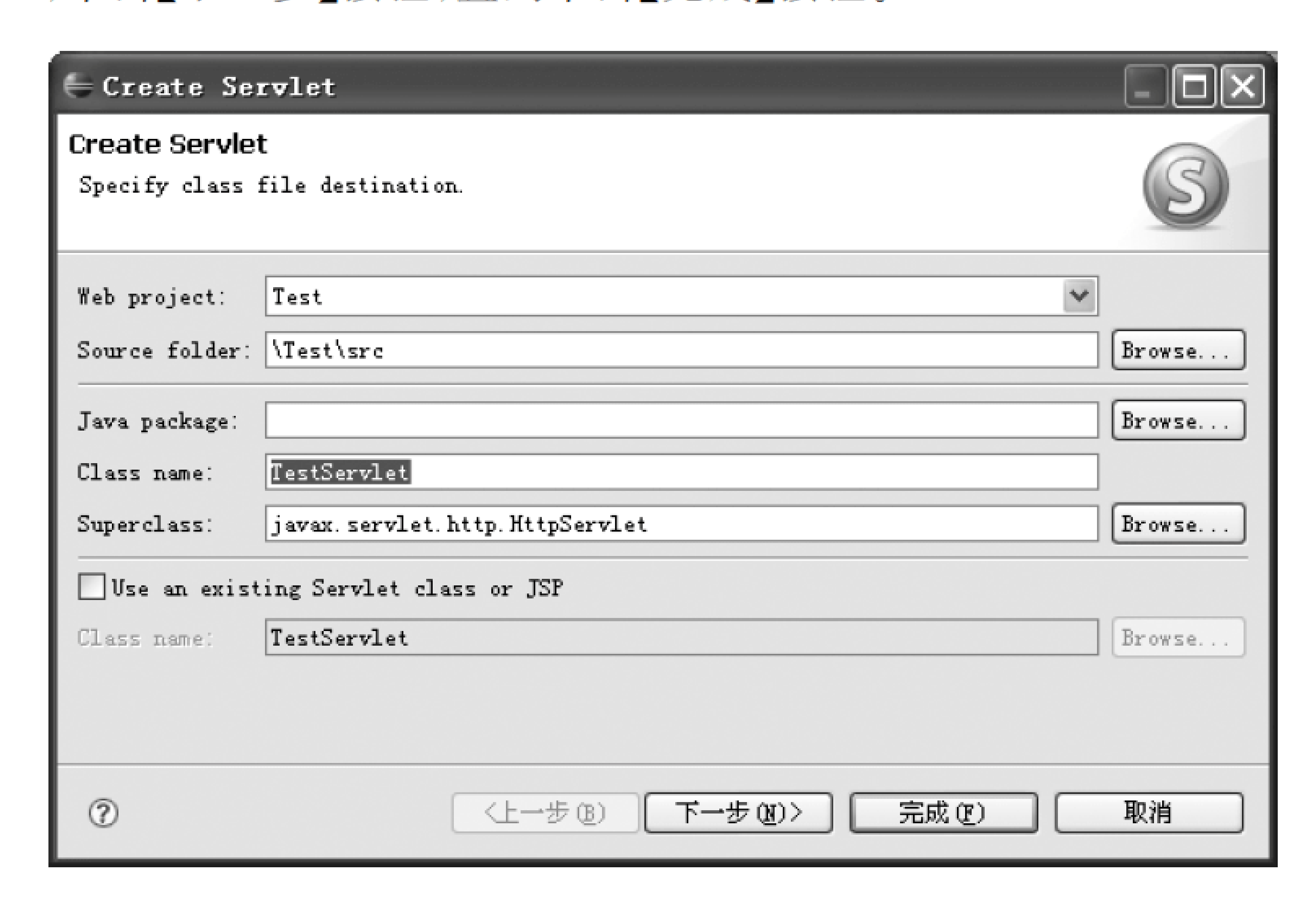


图 5-69　【Create Servlet】对话框

（3）如图 5-70 所示，Eclipse 自动打开刚刚新建的 Servlet 文件，在右侧的工作区中输入相关的 Servlet 代码并保存。

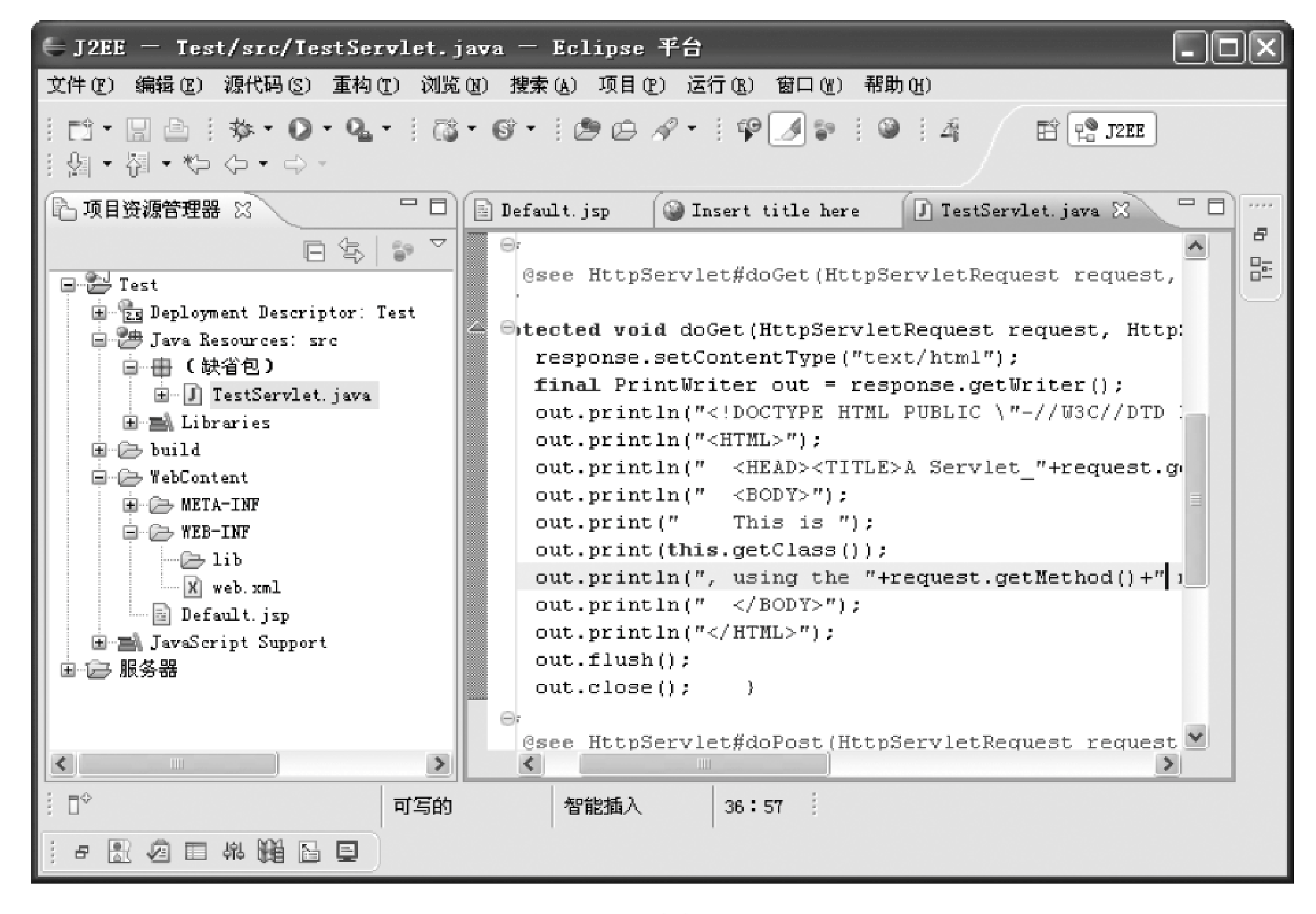


图 5-70　编辑 Servlet



（4）在 Eclipse 界面左侧的【项目资源管理器】中右击并选择新建的 Servlet 文件，在快捷菜单中依次选择【运行方式】|【在服务器上运行】命令，如图 5-71 所示。

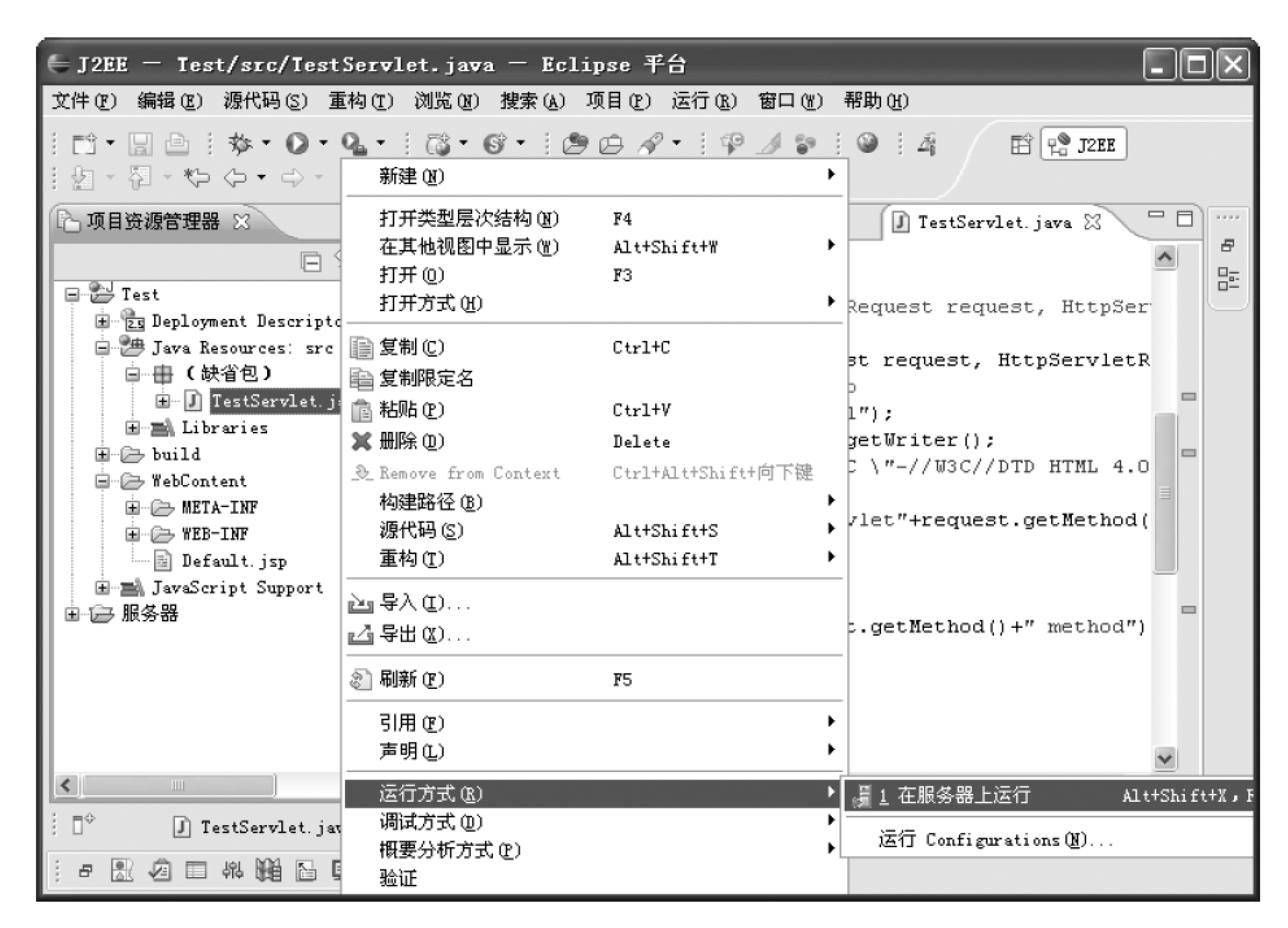


图 5-71　对 Servlet 进行运行方式的选择

（5）为当前运行的 Servlet 文件配置服务器运行时，由于前面运行 JSP 时已经配置了服务器运行时，在此只需选择一个现有的服务器就可以了，如图 5-72 所示。



图 5-72　选择一个现有的服务器





（6）Eclipse 运行服务器并显示运行结果，如图 5-73 所示。

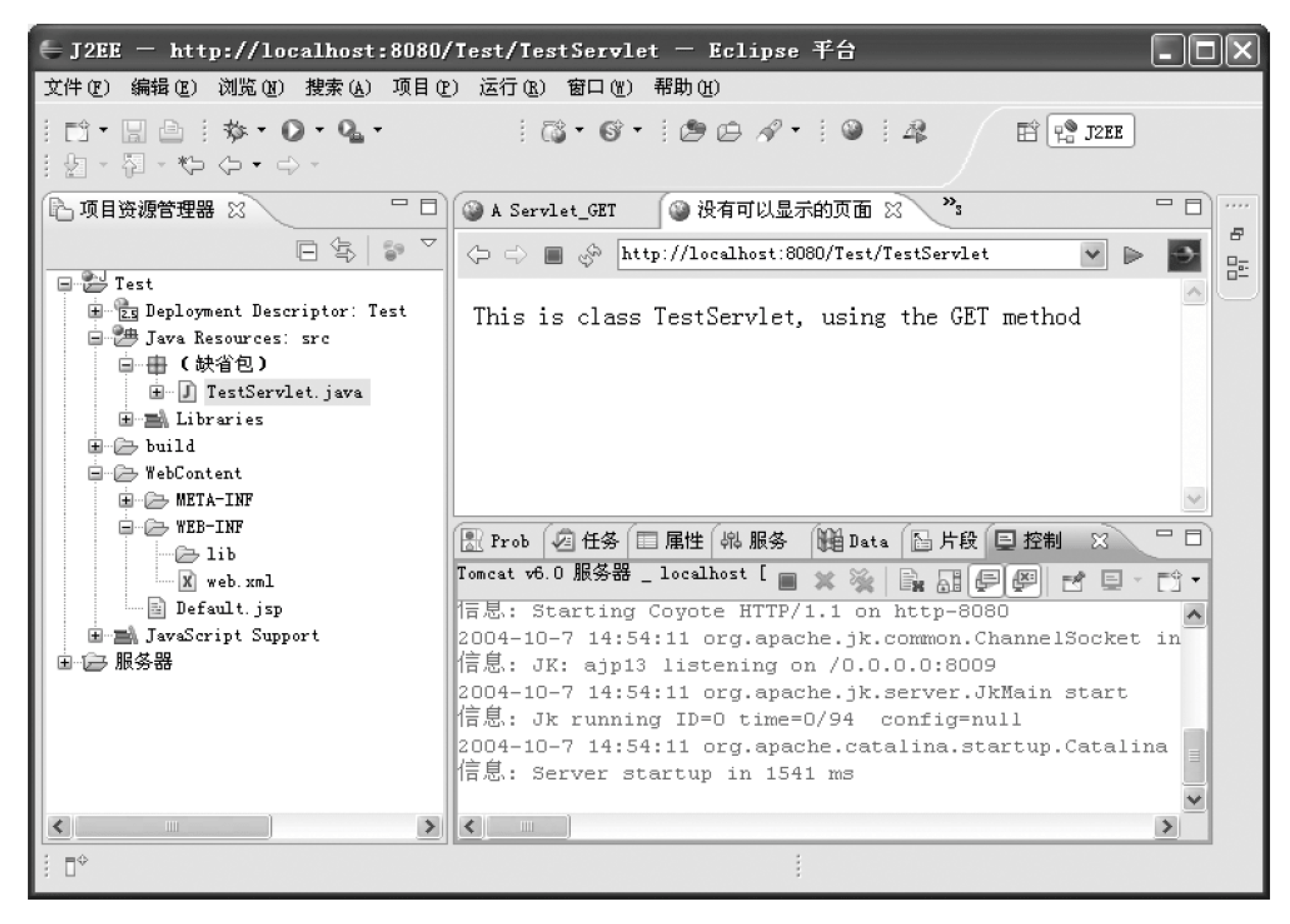


图 5-73　Servlet 运行后的结果

5.6.3　相关知识拓展

Eclipse 未来的目标不仅仅是成为专门开发 Java 程序的 IDE 环境。根据 Eclipse 的体系结构，通过开发插件，它能扩展到支持任何软件开发语言，甚至能成为图片绘制的工具。目前，Eclipse 已经开始提供 C 语言开发的功能插件。更难能可贵的是，Eclipse 是一个开放源代码的项目，任何人都可以下载 Eclipse 的源代码，并可以在此基础上开发自己的功能插件。也就是说未来只要有人需要，就会有建立在 Eclipse 之上的 COBOL、Perl、Python 等语言的开发插件出现。同时可以通过开发新的插件扩展现有插件的功能，比如可以在现有的 Java 开发环境中加入 Tomcat 服务器插件。Eclipse 可以无限扩展，而且有着统一的外观、操作和系统资源管理，这正是 Eclipse 的潜力所在。值得一提的是，Eclipse 项目的参与者除了 IBM 以外，还有像 Borland、Rational Software、RedHat 和 Merant 等一大批业界的佼佼者，这也为 Eclipse 的未来奠定了良好的基础。

5.6.4　经验总结

1．workspace 的优点之一

所有使用 Eclipse 的人最先接触的就是 workspace，因为运行 Eclipse 最先就要设定



workspace 的位置，那么什么是 Eclipse 的 workspace 呢？ workspace 的好处又是什么呢？下面先看一个例子。

打开 Eclipse，新建一个 Java 的 Project（项目）。单击下一步，输入 Test01 作为工程名称，Contents 的配置默认是：Create new project in workspace，此处不用修改，直接单击【完成】按钮。

再新建一个 Java 的 Project。单击【下一步】按钮，输入 Test02 作为工程名称，这一次修改 Contents 的配置为：Create project from existing source，任意输入一个想存放工程的目录（例如 C：\Test02），然后单击【完成】按钮。

关闭 Eclipse，将 workspace 和 Test02 的工程文件夹都移动到其他位置，然后再次打开 Eclipse，选择 workspace 的时候浏览到移动后的 workspace，看看 Test01 工程和 Test02 工程，是不是发现 Test01 还是正常的，而 Test02 却不能打开了？ 也就是说放置在 workspace 中的工程是用相对路径保存工程信息，不论如何移动 workspace 都能正常工作。

很多人都使用共享工程文件夹的方式来共享工程，而每个人放置工程的文件夹都不同，每个人都需要修改工程信息才能工作，又或者想整理自己的工程需要移动工程的位置，处理起来非常麻烦，都是因为没有理解 workspae 的好处。

2．workspace 的优点之二

新建一个 General 的 Project，名称为 jars。然后在这个工程里面放置几个工程公用的 jar 文件（例如 xxx.jar）。假如前面的 Test01 工程需要导入 xxx.jar 这个包，那么单击 Test01 工程的【Properties】，选择【Java Build Path】|【Librares】，再单击【Add JARs】，选择“jars”目录里面的“xxx.jar”。同样的道理，这种方式引用的 jar 也是使用的 workspace 相对路径，以后不论怎么移动 workspace 文件夹，工程配置都不用变化。

Eclipse 中有很多配置都是可以使用 workspace 路径的。

5.7　实　　训

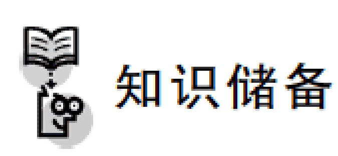
按照本章所述内容，为“新闻类别网站”配置好开发环境。



第6章　JSP应知应会

JSP是Java Server pages的缩写，是Web动态网站开发的主流技术。动态网站技术有CGI、ASP、PHP、JSP等多种，而JSP基于Java语言，具有跨平台性、可重用性、运行效率高、上手容易等优点，因此备受用户和软件开发者的青睐。本章主要介绍JSP的基本知识，包括JSP页面结构、Java语言、JSP语法、JSP内置对象等。

6.1　JSP的基本结构



知识储备

JSP页面程序是在HTML代码之中嵌入Java代码，所以在JSP页面中包括两种主要内容：一是HTML语言，是静态的内容；二是Java程序及其相关元素，是动态的内容。

Java程序及其相关元素主要包括Java脚本元素、JSP指令和JSP动作。Java脚本元素包括JSP表达式、变量和方法的声明以及Java语句。JSP指令用于告诉JSP的Web引擎（如Tomcat）需要在编译时执行的操作，如引入其他的类，设置页面编码方式等。JSP动作是JSP页面被请求时动态执行的操作，如根据某个条件跳转到另外一个页面等。

在HTML中插入Java程序代码，需要把代码写在“＜％”和“％＞”之间。JSP文件的扩展名为.jsp。JSP文件要放到Web服务器上，然后通过HTTP的方式访问。

JSP页面被访问时，页面中动态的内容将由服务器负责执行，并把执行结果转变为字符串，然后把这些结果和静态内容一起交给客户端的浏览器显示出来。

为了增强程序的可读性，可以在JSP程序中加入注释。JSP程序中的注释可分为两种：HTML注释和JSP注释。

1．HTML注释

放在符号“＜！--”与“--＞”之间的内容是HTML注释，其格式如下：

```
<!--注释内容-->
```

JSP不解释HTML注释，而是直接把它交给客户端的浏览器。因此，这种注释在浏览器中直接查看JSP源文件时可以看到。



2. JSP 注释

JSP 注释的方法有以下四种,格式分别如下:

```
<%--注释内容--%>
<%//注释内容%>
<%/*注释内容*/%>
<%/**注释内容*/%>
```

第一种注释称为隐藏注释,写在 JSP 程序中,但并不发送给客户端。后三种是 Java 语言的注释方式,分别为单行注释、多行注释和 doc 注释。

6.1.1 任务描述

实现第一个简单的 JSP 页面,在浏览器上显示"用户登录成功"的信息。

6.1.2 任务实现

具体操作步骤如下:

(1) 新建项目。打开 Eclipse,新建 Web 项目 Test。

(2) 新建 JSP 文件。选中项目 Test,新建 JSP 文件,文件名为 admin_success.jsp。

(3) 编辑 JSP 文件。在该文件的编辑窗口中,输入如程序 6-1 所示的代码,保存文件。

【程序 6-1】 admin_success.jsp 文件代码

Example Source Code:

```
<%@page contentType="text/html;charset=gb2312"%>
<html>
<head>
<title>登录成功页面</title>
</head>
<body>
<%
//向浏览器输出要显示的内容
  out.println("用户登录成功");
%>
</body>
</html>
```

(4) 部署 Web 项目。选中项目 Test,部署到 Tomcat 的 webapps 目录下,并启动 Tomcat 服务器。

(5) 浏览 JSP 网页。打开 IE 浏览器,在地址栏中输入 http://localhost:8080/Test/admin_success.jsp,则可以看到运行结果,如图 6-1 所示。

代码说明:

JSP 页面中包括<html>、<head>、<title>和<body>等 HTML 标记。

Java 程序代码用"<%"和"%>"括起来,out.println 语句用来向浏览器输出要显示的





图 6-1　登录成功页面

内容，“//”注释是单行注释。

为了在 JSP 页面中支持中文显示，在开头加了如下语句：

```
<%@page contentType="text/html;charset=gb2312"%>
```

6.1.3　相关知识拓展

常见的网络计算模式有 C/S 模式和 B/S 模式。

C/S(Client/Server)是客户/服务器模式，其工作分别由服务器和客户机完成。服务器负责管理数据库的访问，通常采用高性能的 PC、工作站或小型机，并采用大型数据库系统，如 Oracle、Sybase、Informix 或 SQL Server。客户机需要安装专用的客户端软件，负责与用户的交互和对服务器的请求。最简单的 C/S 模式应用由两部分组成，客户应用程序和数据库服务器程序，又分别称为前台程序和后台程序。

B/S(Brower/Server)是浏览器/服务器模式，客户机上只需要安装浏览器，如 Netscape 或 IE，通过浏览器向 Web 服务器发出请求，由 Web 服务器对数据库进行操作，并将结果返回给客户端。B/S 模式简化了客户机的工作，事务逻辑处理全部在服务器端进行。

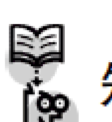
相比较而言，C/S 模式的优点是能充分发挥客户端 PC 的处理能力，响应速度快，安全性高；而 B/S 模式的优点是客户端无须安装专门的软件，易维护、易扩展。随着 Internet 逐步走进人们的日常生活，客户简便化的使用要求使得 B/S 比 C/S 更具有发展优势。JSP 就是 B/S 模式下的一种开发技术。

6.1.4　经验总结

程序员在程序中添加注释是一个良好的习惯，在平时学习和实践时要注意培养这一习惯。注释可以提高程序的可读性，方便程序编写，方便相互交流。另外，注释在调试程序中也特别有用，比如在调试一段程序的其中几行程序时，可以把其他的程序放入注释中，这样 JSP 引擎就不会编译它，使程序只运行正在调试的程序段，提高暴露错误的概率，这是一种很常用的程序调试方法。



6.2　Java 语言介绍


知识储备

要学好 JSP 程序开发，必须了解 Java 语言。Java 语言是一种面向对象的语言，是 JSP 的脚本编程语言。下面介绍常用的 Java 编程基础知识。

1. 数据类型

Java 的数据类型可分为两种：简单数据类型和引用数据类型。简单数据类型包括整型、实型、布尔型、字符型，引用数据类型包括数组、类、接口。

Java 的简单数据类型如表 6-1 所示。

表 6-1　Java 的简单数据类型

简单数据类型	类型含义	所占空间	简单数据类型	类型含义	所占空间
byte	字节型	1 字节	boolean	布尔型	1 位
short	短整型	2 字节	char	字符型	2 字节
int	整型	4 字节	float	单精度实型	4 字节
long	长整型	8 字节	double	双精度实型	8 字节

(1) 常量与变量

常量是在程序运行过程中其值不变的量。常量区分为不同类型，如整型常量 12，实型常量 12.3，字符常量'a'，布尔常量 true，字符串常量"Hello"。

变量是用来存放指定类型数据的量，其值在程序运行过程中是可变的。变量声明的一般格式如下：

```
<数据类型>　<变量名>=<变量值>　;
```

【例 6-1】　变量定义示例

```
int i=1;
double x=123.45;
boolean b=true;
```

(2) 整型数据

Java 整型数据多为十进制形式，也可为八进制或十六进制形式，八进制以 O 开头，十六进制以 Ox 开头。Java 中整数默认是 int 型，如果为长整型数，需在数据后加字母 l 或 L。

整型变量有 4 种类型：byte、short、int 和 long。最常用的是 int 类型，可满足大多数场合应用的需要，如果需要更大的整数，可以使用 long 类型。

(3) 实型数据

实型数据用十进制形式表示，由数字和小数点组成，如 123.45。也可以用科学计数法形式表示，如 1.2345E2。实型数据默认是 double 类型，加 f 或 F 为 float 类型。

实型变量可分为 float 和 double 两种类型。float 为单精度实型，double 为双精度实型，



double 类型比 float 类型表示数据的精度更高，表示数据的范围更大。

(4) 字符型数据

字符型常量是用单引号括起来的一个字符，如'A'、'好'。

字符型变量为 char 类型，如 char ch='A'。

Java 中有一些字符以反斜杠(\)开头，反斜杠将其后面的字符转变为另外的含义，称为转义字符。常用的转义字符如表 6-2 所示。

表 6-2 Java 的转义字符

转义字符	含　义	转义字符	含　义
\t	Tab(制表)	\'	单引号
\n	换行	\"	双引号
\r	回车	\\	反斜杠
\b	退格		

(5) 布尔型数据

布尔型常量只有两个：true 和 false，分别表示逻辑真和逻辑假。

布尔型变量为 boolean 类型，取值为 true 或 false。

2. 运算符与表达式

运算符可以用来处理数据，运算符与操作数组成的式子称为表达式。常用的运算符有算术运算符、赋值运算符、关系运算符和逻辑运算符等。

(1) 算术运算符

算术运算符完成算术运算功能，包括+、-、*、/、%、++、--等运算符。算术运算符如表 6-3 所示。

表 6-3 算术运算符

运　算　符	名　称	说　明	举　例
+	加法运算符	求两个数的和	5+3 等于 8
-	减法运算符	求两个数的差	5-3 等于 2
*	乘法运算符	求两个数的乘积	5 * 3 等于 15
/	除法运算符	求两个数的商	5/3 等于 1
%	求余运算符	求两个数相除的余数	5%3 等于 2
++	自增运算符	使变量的值自增 1	i++等价于 i=i+1
--	自减运算符	使变量的值自减 1	i--等价于 i=i-1

(2) 赋值运算符

赋值运算符完成给变量提供数据的功能，"="号是赋值运算符。

赋值表达式的一般形式为：变量=表达式。

赋值表达式的处理过程是：先计算"="号右侧表达式的值，然后将该值赋给左侧的变量。



【例 6-2】 赋值运算符示例

```
int a=1;
int b=a+5;
```

(3) 关系运算符

关系运算符完成操作数的比较运算,结果为布尔类型值。关系运算符如表 6-4 所示。

(4) 逻辑运算符

逻辑运算符完成操作数的逻辑运算,要求操作数必须是布尔类型,逻辑运算的结果也是布尔类型值,true 或 false。常用的逻辑运算符如表 6-5 所示。

表 6-4 关系运算符

关系运算符	含 义	关系运算符	含 义
>	大于	<=	小于等于
>=	大于等于	==	等于
<	小于	!=	不等于

表 6-5 逻辑运算符

逻辑运算符	含 义
&&	逻辑与
||	逻辑或
!	逻辑非

3. 分支语句

Java 提供了两种常见的分支控制语句:if 语句和 switch 语句,其中 if 语句使用布尔表达式作为分支条件来进行分支控制,而 switch 语句则用于对多个整型值进行匹配,从而实现分支控制。

(1) if 分支语句

if-else 语句的语法格式如下:

```
if(布尔表达式)
    语句 1;
else   //else 分支可以省略
    语句 2;
```

其中,表达式是判断条件,如果条件为真(true),则执行语句 1;如果条件为假(false),则执行语句 2。

执行流程如图 6-2 所示。

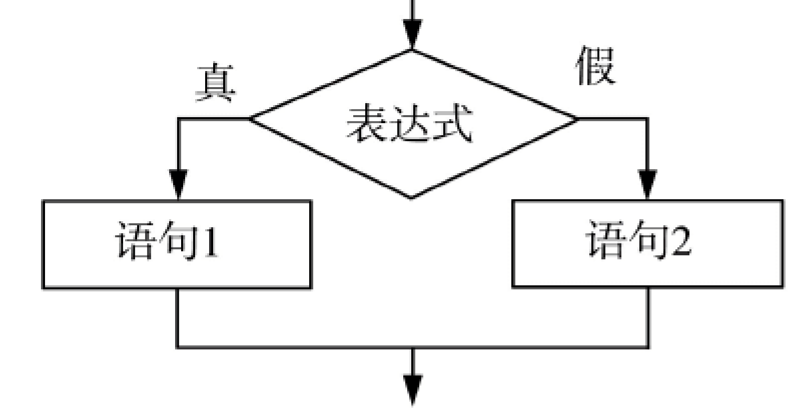


图 6-2 if-else 语句的流程图

【例 6-3】 if 语句示例

```
if(x>y)
    max=x;
else
    max=y;
```

上面代码段的含义是将变量 x 和 y 中较大的一个值赋给变量 max。

if 语句的另一个常用形式是多分支 if 语句。

多分支 if 语句的语法格式:

```
if(表达式 1)
    语句 1;
else if(表达式 2)
```



```
    语句 2;
else if(表达式 3)
    语句 3;
    ⋮
else if(表达式 n)
    语句 n;
else
    语句 n+1;
```

多分支 if 语句是在 if 语句的 else 部分中包含其他 if 块，其执行过程为：依次判断表达式的值，当出现某个值为真(true)时，则执行其对应的语句，然后跳到整个 if 语句之外继续执行程序。如果所有的表达式均为假(false)，则执行语句 n+1，然后继续执行后续程序。多分支 if 语句可以有零个或多个 else if。

多分支 if 语句的执行流程如图 6-3 所示。

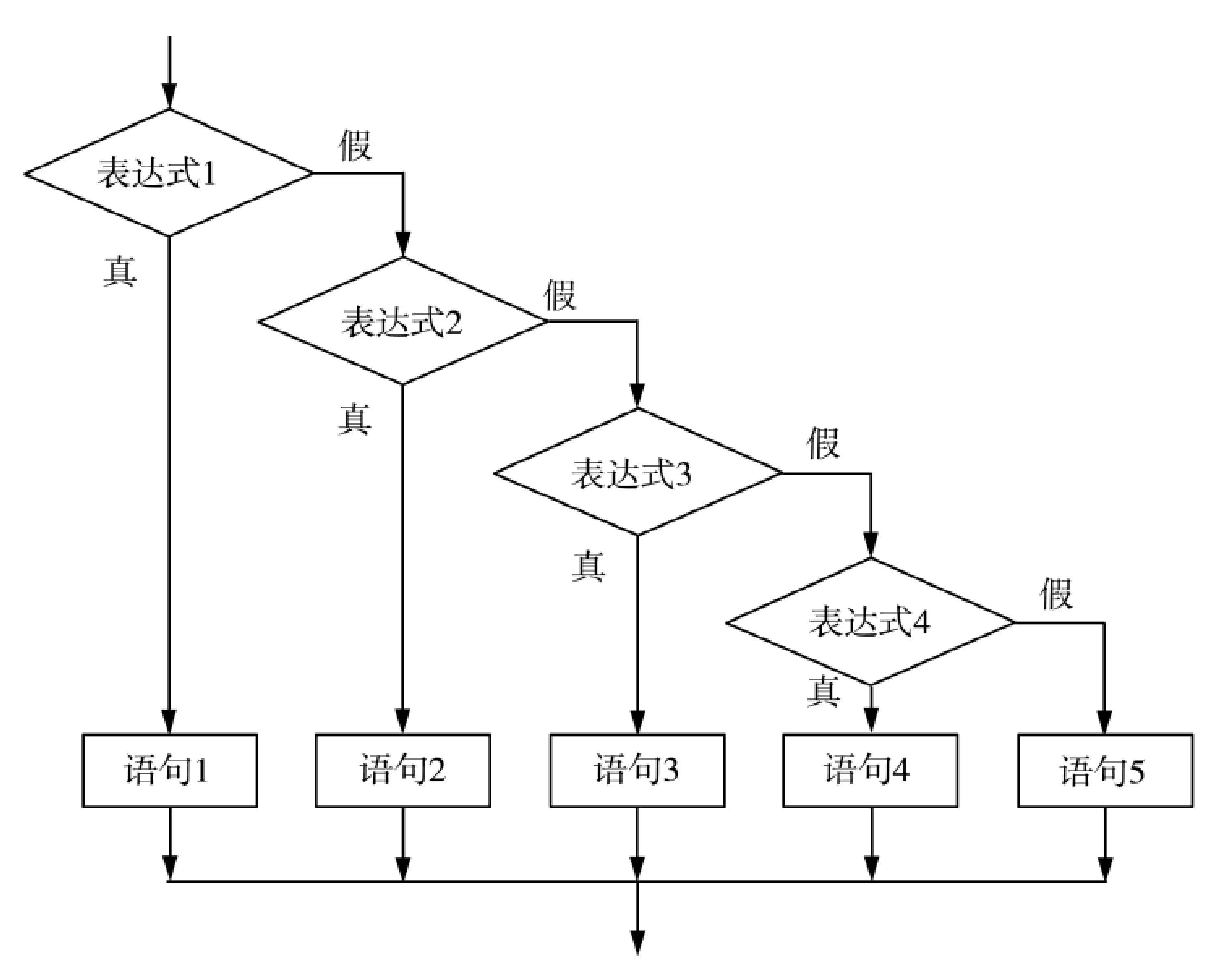


图 6-3　多分支 if 语句的流程图

【例 6-4】　多分支 if 语句示例，如程序 6-2 所示。

【程序 6-2】　多分支 if 语句示例代码

Example Source Code：

```
public class TestIf{
  public static void main(String[] args){
    int score=85;
    if(score>=90)
      System.out.println("优秀");
    else if(score>=80)
      System.out.println("良好");
    else if(score>=70)
      System.out.println("中等");
    else if(score>=60)
```



```
        System.out.println("及格");
      else
        System.out.println("不及格");
      }
}
```

上面程序保存成 Testlf.java 文件，其运行结果是输出"良好"。Java 源程序以.java 为扩展名，编译生成.class 文件。

(2) switch 语句

switch 语句的语法格式如下：

```
switch(表达式){
    case 常量 1: 语句 1;   break;
    case 常量 2: 语句 2;   break;
      ⋮
    case 常量 n: 语句 n;   break;
    default:       语句 n+1;   break;
}
```

switch 后面的表达式的数据类型只能是整型或字符型。

switch 语句的执行流程为：先计算表达式的值，然后依次与 case 后面的常量相比较，如果表达式的值与某一个 case 的常量值相同时，则执行该 case 后面的语句，然后通过 break 跳出 switch 语句。如果表达式的值与所有 case 后面的常量均不相同时，则执行 default 后面的语句。default 子句可以省略不写。

【例 6-5】 switch 语句示例，如程序 6-3 所示。

【程序 6-3】 switch 语句示例代码

Example Source Code:

```
public class TestSwitch{
  public static void main(String[] args){
      char grade='C';
      switch ( grade ) {
          case 'A': System.out.println("优秀"); break;
          case 'B': System.out.println("良好"); break;
          case 'C': System.out.println("中等"); break;
          case 'D': System.out.println("及格"); break;
          case 'E': System.out.println("不及格"); break;
          default: System.out.println("成绩输入错误");
      }
}
```

上面程序的运行结果是输出"中等"。

4. 类和对象

Java 程序由一些类组成。类的定义要用关键字 class，对象的创建要用关键字 new。

类的定义格式如下：

```
[public] class 类名 {
    类体
```



```
}
```

其中，类体包括变量和方法两部分，分别用来描述对象的状态和行为。

创建对象格式：

```
类名　对象名=new 类名();
```

【例 6-6】 类与对象示例，如程序 6-4 所示。

【程序 6-4】 类与对象示例代码

Example Source Code：

```
public class Manager{
    private String name;                              //成员变量
    public Manager(String n)                          //构造方法
    {
        name= n;
    }
    public void display()                             //display 方法
    {
        System.out.println("用户名："+name);
    }
    public static void main(String[] args)            //main 方法
    {
        Manager manager= new Manager("admin");        //创建对象
      manager.display();                              //调用 display 方法
    }
}
```

上述代码定义了一个 Manager 类，类中定义了一个属性变量 name，定义了三个方法，分别为构造方法、display( )方法和 main( )方法。构造方法的方法名与类名相同，在创建对象时会自动调用构造方法进行一些初始化工作。display( )方法是自定义方法，该方法的功能是显示用户信息。main( )方法是程序的入口。一个完整的 Java 程序可以由多个文件组成，主程序应包括一个 main( )方法，解释器正是从该方法开始执行 Java 程序，main 方法一般具有如下格式：

```
public static void main(String[] args){
  //方法体
}
```

例 6-6 程序的运行结果为：

```
用户名：admin
```

Java 系统类库中提供了许多系统类，String 类是一个系统类，代表字符串类型。String 类有很多字符串处理方法，其中的 equals( )方法很常用，用来判断两个字符串是否相同。在本节的任务实现时会用到 String 类型和 equals( )方法。

6.2.1　任务描述

实现一个 JSP 页面，页面中包含进行用户名和密码验证的 Java 程序代码。



6.2.2 任务实现

操作步骤同 6.1 节，新建文件 admin_check.jsp，代码如程序 6-5 所示。

【程序 6-5】 admin_check.jsp 文件代码

Example Source Code：

```
<html>
<head>
<title> 登录验证页面</title>
</head>
<body>
<% String username= "admin ";
    String password;
    password= "000000";
  if( username.equals("admin") && password.equals("000000") )
        out.println("用户信息匹配");
  else
        out.println("用户信息不匹配");
%>
</body>
</html>
```

代码说明：

插入的 Java 代码中，定义了两个字符串变量 username 和 password 并赋初值，然后用 if 语句判断，如果 username 的值为 admin，并且 password 的值为 000000，则在网页上输出用户信息匹配，否则输出用户信息不匹配。因为变量的初始化值和比较字符串相同，所以网页上显示的是用户信息匹配。

程序的运行结果如图 6-4 所示。

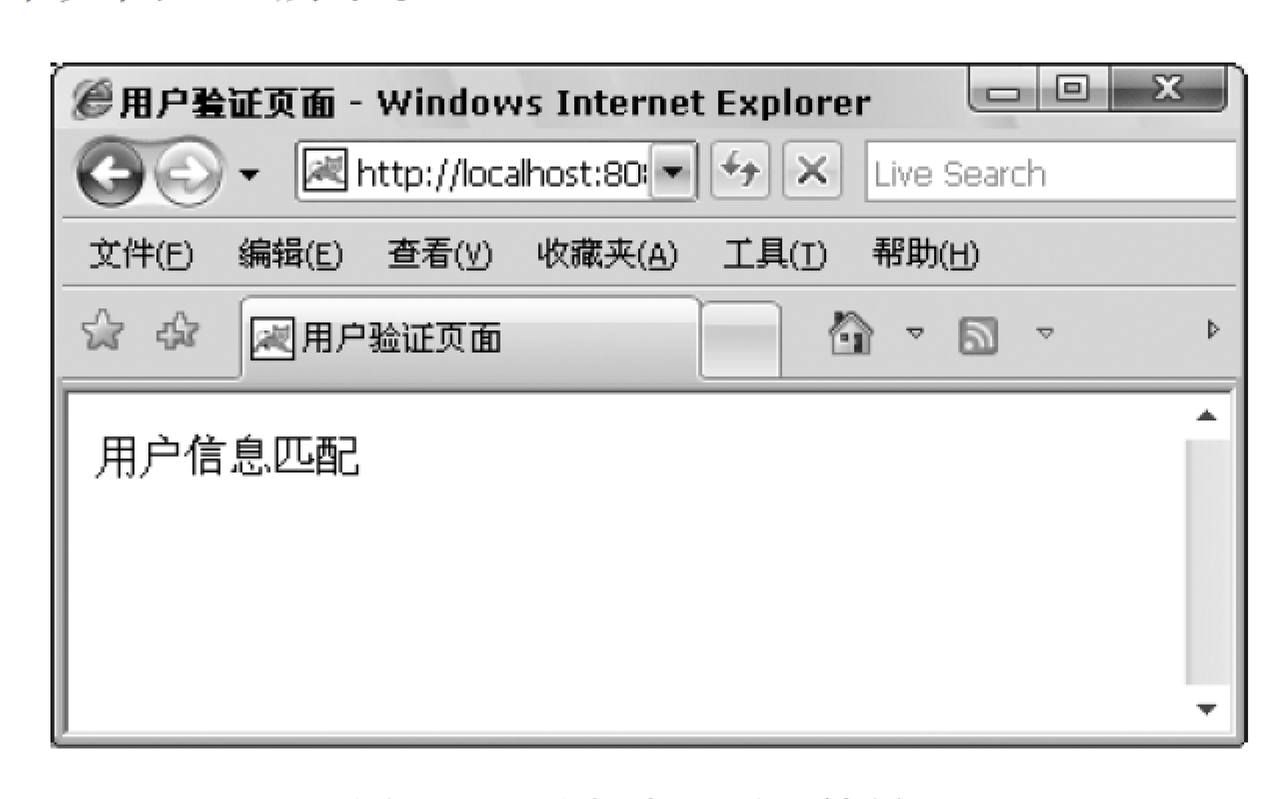


图 6-4 用户验证页面结果 1

修改上述代码中 password="123456"，则 JSP 页面的浏览效果如图 6-5 所示。

小贴士

实际应用时，这段示例代码并不能进行真正意义上的用户身份验证。

真正进行身份验证时，是对访问者在网页上输入并提交给服务器的信息进行验证。输



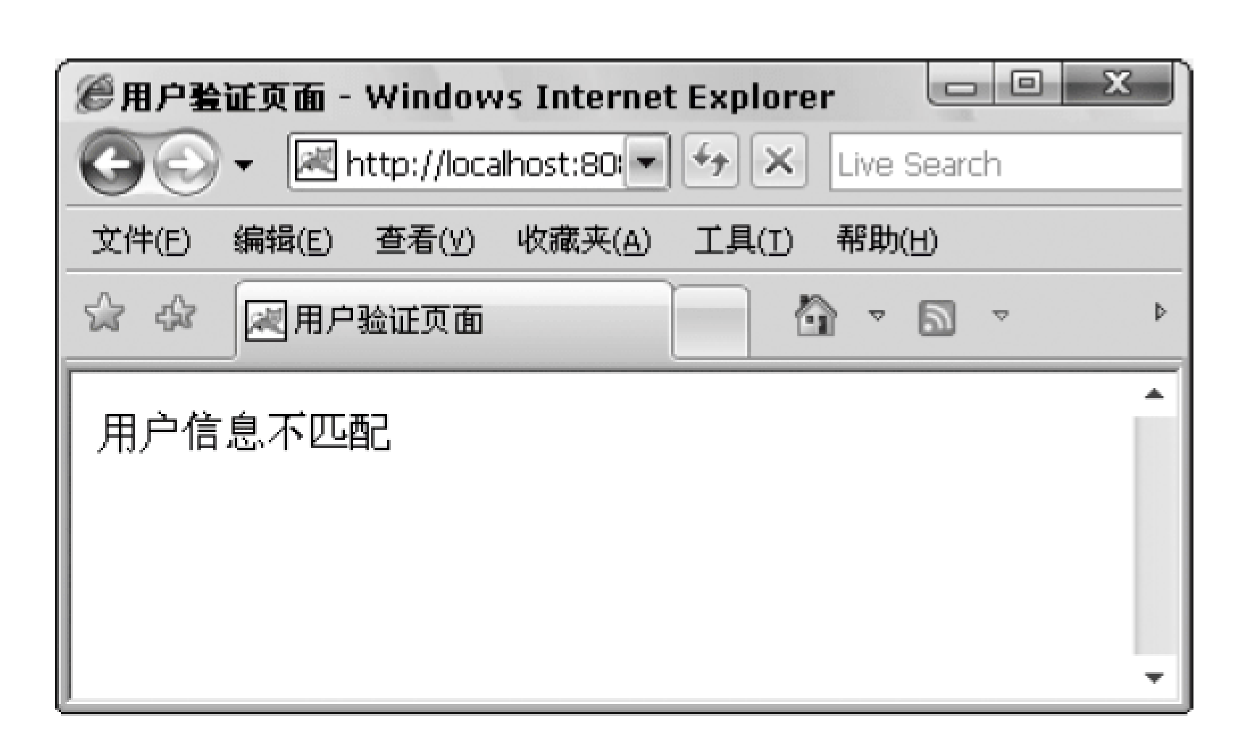


图 6-5　用户验证页面结果 2

入和提交信息通过表单来实现，可通过 JSP 的 request 对象获得由浏览器提交的信息。关于 request 对象，会在 6.4 节中介绍。正确的用户名和密码存放在服务器的数据库中，身份验证时用数据库中的信息与浏览器提交的访问者信息进行比较，以进行身份的合法性检查，这些内容将在后续章节中继续学习。

6.2.3　相关知识拓展

1. 程序控制逻辑

在 Java 中控制逻辑分为分支、循环和跳转三种，分支语句在前面已经介绍过，这里不再赘述。

(1) 循环语句

循环是指在满足循环条件的情况下，反复执行某一段代码，这段被重复执行的代码称为循环体。循环语句可能包含如下四个部分。

- 初始化语句：一条或多条语句，用于完成一些初始化工作，在循环开始之前执行。
- 循环条件：是一个布尔类型表达式，由这个表达式决定是否执行循环体。
- 循环体语句：是循环的主体，如果循环条件为真，这个代码块将被重复执行。
- 循环变量变化语句：在一次循环体执行结束之后，循环条件求值之前执行该语句，通常用于控制循环条件中的变量，使得循环在合适时候能够结束。

Java 中的循环语句有三个：while 语句、do-while 语句和 for 语句。

① while 语句

while 循环的语法格式如下：

```
while(循环条件){
    循环体
}
```

【例 6-7】　while 循环示例，如程序 6-6 所示。

【程序 6-6】　while 循环示例代码

Example Source Code:

```
int i=1;
```



```
while(i<=100){
    System.out.println("我能行");
    i++;
}
```

② do-while 循环

do-while 循环的语法格式如下：

```
do{
    循环体
} while(循环条件);
```

【例 6-8】 do-while 循环示例，如程序 6-7 所示。

【程序 6-7】 do-while 循环示例代码

Example Source Code：

```
int i=1;
do{
    System.out.println("我能行");
    i++;
} while(i<=100);
```

③ for 循环

for 循环的语法格式如下：

```
for(表达式 1; 表达式 2; 表达式 3)
    循环体
}
```

其中，表达式 1 为循环变量初始化，表达式 2 为循环条件，表达式 3 为循环变量变化。

【例 6-9】 for 循环示例，如程序 6-8 所示。

【程序 6-8】 for 循环示例代码

Example Source Code：

```
for(int i=1; i<=100; i++){
    System.out.println("我能行");
}
```

（2）跳转语句

① break 语句

有些时候，需要在某种条件出现时，强行终止循环，而不是等到循环条件为 false 时。此时，可以使用 break 来完成这个功能。break 用于完全结束一个循环，跳出循环体，开始执行循环之后的代码。

【例 6-10】 break 语句示例，如程序 6-9 所示。

【程序 6-9】 break 语句示例代码

Example Source Code：

```
for(int i=0; i<5; i++){
```



```
    if(i==2) break;
    System.out.println("i="+i );
}
```

上面程序段中,当 i 等于 2 时,执行 break 语句,能使整个循环强制退出。

运行结果:

```
i=0
i=1
```

② continue 语句

continue 和 break 的区别是 continue 只是中止本次循环,接着开始下一次循环,而 break 则是完全终止循环。可以理解为 continue 的作用是略过当次循环中剩下的语句,重新开始新的循环。

【例 6-11】 continue 语句示例,如程序 6-10 所示。

【程序 6-10】 continue 语句示例代码

Example Source Code:

```
for(int i=0;i<5;i++){
    if(i==2) continue;
    System.out.println("i="+i );
}
```

上面程序段中,当 i 等于 2 时,提前结束本次循环,未执行输出语句,进入下一次循环。

运行结果:

```
i=0
i=1
i=3
i=4
```

③ return 语句

return 的功能是结束一个方法,return 后可以跟变量、常量和表达式,把值返回给上一级调用方法处。当一个方法执行到一个 return 语句时,这个方法将被结束。

2. 数组

数组的作用是存储多个数据类型相同的数据,其中的一个数据称为数组元素,通常可通过数组元素的索引来访问数组元素。数组也是一种数据类型,它本身是一种引用类型。例如 int 是一个基本类型,但 int[](这是声明数组的一种方式)就是一种引用类型了。

(1) 声明数组

Java 支持下面两种数组声明形式,一般推荐使用第一种。

```
数据类型[] 数组名;
数据类型 数组名[];
```

例如:

```
int[] a;
```



声明 a 是一个整型数组。

数组是一种引用类型的变量，因此使用它声明一个变量时，仅仅表示定义了一个引用变量（也就是定义了一个指针），这个引用变量还未指向任何有效的内存空间，因此声明数组时不能指定数组的长度，而且也没有分配内存空间来存储数组元素。

（2）分配空间

给数组分配空间使用 new 关键字，语法形式如下：

```
数组名=new 数据类型[数组长度];
```

例如：

```
a=new int[10];
```

也可以在声明数组的同时指定数组长度，格式为：

```
数据类型[] 数组名=new 数据类型[数组长度];
```

例如：

```
int[] a=new int[10];
```

（3）数组初始化

数组初始化是为数组元素赋初值。用 new 关键字为一个数组分配空间后，系统将为数组元素赋 0 值或空值。可以用赋值语句给数组的每个元素初始化。数组元素通过“数组名[下标]”来引用。

例如：

```
a[0]=1;
```

也可以在声明数组的同时进行数组初始化，这时不需要用 new 关键字。

例如：

```
int a[]={1,2,3,4,5};
```

（4）使用数组

数组的主要使用方法就是访问数组元素，包括对数组元素进行赋值和访问数组元素的值，数组元素的引用方式为“数组名[下标]”。值得指出的是，Java 语言的数组下标是从 0 开始的，也就是说，第一个数组元素的下标为 0，最后一个数组元素的下标为数组长度减 1。

所有数组都提供了一个 length 属性，通过这个属性可以访问到数组的长度。

【例 6-12】 数组示例，如程序 6-11 所示。

【程序 6-11】 数组示例代码

Example Source Code：

```
int[] a={1,2,3,4,5};
for(int i=0; i<a.length; i++)
{
    System.out.println(a[i]);
}
```



该代码段的功能是给 a 数组赋值，并输出 a 数组每个元素的值。

3．类的继承

类之间可以有继承关系，被继承的类称为父类，继承父类的类称为子类。执行继承时，子类可以获得父类的属性和操作，并具有自身特有的属性和操作。

extends 关键字用于继承类，声明一个继承父类的类的通常形式如下：

```
class 子类名 extends 父类名{
        //子类代码
}
```

如果类的声明中没有 extends，则默认父类为 Object。Java 语言中子类只能有一个父类，具有单继承性，要用多继承时，可以通过下面介绍的接口来实现。

4．接口

接口是面向对象的程序设计中的一个重要概念，利用接口可以使设计与实现相分离。Java 中只允许单继承，可以利用接口实现多继承。一个接口可以继承多个接口，一个类可以实现多个接口。

（1）接口的声明

接口的声明格式如下：

```
[public] interface 接口名称{
    [public] [static] [final] 变量名=初值;              //静态常量
    [public] [abstract] 返回值类型 方法名([参数表])
}
```

由于接口中声明的方法都是抽象的、公有的，声明的常量都是静态的、公有的，所以 public、static、final、abstract 等修饰符都可以省略。

（2）接口的实现

在类的声明中使用 implements 关键字来表示实现接口，一个类可以同时实现多个接口，各接口间用“，”隔开。

```
class classname implements interfacename{
    //重写接口中的方法
}
```

若实现接口的类不是抽象类，则必须实现接口的所有方法，即为所有的抽象方法声明方法体。一个类在实现某接口的抽象方法时，必须使用相同的方法名和参数列表。

5．异常处理

异常是在程序的运行过程中所发生的不正常的情况，它会中断正在运行的程序。Java 语言使用异常处理机制为程序提供了错误处理的能力。

Java 语言使用 try-catch-finally 块捕获并处理异常，语法形式为：

```
try{
```



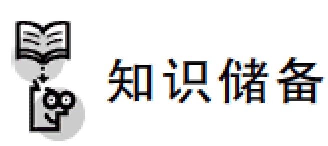
```
        //可能发生异常的代码
    }
    catch(异常类型 e){
        //对异常进行处理的代码
    }
    finally{
        //总要被执行的代码
    }
```

当 try 块中的代码发生异常时,会产生异常对象。异常是一种特殊的对象,当异常对象的类型与 catch 后的异常类型匹配时,可以捕获到该异常,此时 catch 块中的异常处理代码将执行。而 finally 中的代码无论是否发生异常都会执行。

6.2.4 经验总结

Java 语言是学习开发 JSP 网站的必备基本功。在用 JSP 技术开发网站时,一定会用到 Java 语言知识和 Java 编程思想。因此,既要掌握 Java 语法,还要理解 Java 面向对象的思想,要学会分析实际问题,合理设计类,这将有助于更深入地学习 JSP 技术。

6.3 JSP 语法



知识储备

JSP 语法是建立在 Java 语言基础上的,但它是一种 Web 程序设计语言,故有自己的一些特有的语法。

1. JSP 指令

"<%@"和"%>"之间为 JSP 指令,常用的有两种:第一种是 page 指令,用来完成导入指定的类,或指定页面编码方式等;第二种是 include 指令,用来包含文件。

(1) page 指令

<%@page %> 指令作用于整个 JSP 页面,可以用于定义 JSP 页面的全局属性,一般将它放在页面的顶部。

【例 6-13】 page 指令示例

```
<%@page contentType="text/html;charset=gb2312" %>
```

页面代码开头有了此句后,在 JSP 页面中就可以正常显示中文字符了。contentType 属性用于定义设置 JSP 文件和最终文件的 MIME 类型和字符集类型,这一项必须放在文件的顶部。contentType 属性的默认值为"text/html; charset=ISO8859_1",为了让 JSP 页面可以正常地显示中文,需要将 contentType 属性设置为"text/html;charset=GBK"或者设置为"text/html;charset=gb2312"。

指令中的属性值可用单引号或双引号括起来,如果一个属性有多个值就用逗号隔开,在 page 指令中也只有 import 属性可以指定多个值,它用来导入一些程序中要用到的包或类,如:

```
<%@page import="java.util.* , java.io.* "%>
```



也可以将需要引用的多个包和类写在多个 page 指令中，例如，上面的语句等价于：

```
<%@page import="java.util.*"%>
<%@page import="java.io.*"%>
```

当使用 import 引入了包和类后，就可以使用与这些包和类相关的方法和对象了。

page 指令的 language 属性指定页面使用哪种编程语言作为脚本语言，默认值为 Java，因此在编写 JSP 程序时可以不设置。

（2）include 指令

include 指令用来在 JSP 文件中包含一个静态的文件，即在 JSP 文件的该指令处插入另一文件的文本或代码，以组成一个更大的程序文件。

include 指令语法格式如下：

```
<%@include file="文件路径"%>
```

当使用<%@include%>指令时，包含的过程是静态的。静态的包含就是指这个被包含的文件的内容将会被导入到 JSP 文件中，这个被包含的文件可以是 JSP 文件、HTML 文件、文本文件。如果被包含的是 JSP 文件，这个被包含的 JSP 文件的代码将会被执行。

【例 6-14】 include 指令示例

① 新建文本文件 top.txt，代码如程序 6-12 所示。

【程序 6-12】 top.txt 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312" %>
<html>
<head>
<title>新闻类别网站</title>
</head>
<body>
<h1>新闻类别网站</h1>
```

② 新建文本文件 bottom.txt，代码如程序 6-13 所示。

【程序 6-13】 bottom.txt 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312" %>
<hr>
<p align="center">
版权信息
</p>
</body>
</html>
```

③ 新建 JSP 文件 includeTest.jsp，代码如程序 6-14 所示。

【程序 6-14】 includeTest.jsp 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312" %>
```



```
<%@include file="top.txt" %>
<p align="center">
<%out.println("快来学习如何开发 JSP 网站吧!");
%>
</p>
<%@include file="bottom.txt" %>
```

上面 3 个文件中，top.txt 中是文件头的代码，bottom.txt 中是文件尾的代码，主文件 includeTest.jsp 中，先包含 top.txt，然后输出一句话，又包含 bottom.txt。

网页运行结果如图 6-6 所示。



图 6-6 include 指令示例的运行结果

在同一个网站中，许多网页的头部和尾部是一样的，因此可以把头部和尾部各自保存为一个文件，在其他文件中只要将其包含进来即可，这样可以大大减少网页代码编写的工作量，同时又能保证网页风格的统一。

2. JSP 动作指令

JSP 动作指令与 JSP 指令不同，可以用它来控制 JSP 引擎的行为。下面介绍 include 动作指令、forward 动作指令和 param 动作指令。

(1) include 动作指令

include 动作指令用来在 JSP 页面中动态包含一个文件。

include 动作指令的格式如下：

```
<jsp: include page="文件路径 "/>
```

或者：

```
<jsp: include page="文件路径">
<jsp: param name="参数 1" value="参数 1 的值"/>
⋮
<jsp: param name="参数 n" value="参数 n 的值"/>
</jsp: include>
```

在不需要传递参数时，采用第一种格式，如果要传递参数则使用第二种形式。



【例 6-15】　include 动作指令示例

修改例 6-14 的主文件 includeTest.jsp，代码如程序 6-15 所示，其运行结果与例 6-14 一样。

【程序 6-15】　includeTest.jsp 文件代码

Example Source Code：

```
<%@ page contentType="text/html;charset=gb2312" %>
<%jsp:include page="top.txt" %>
<p align="center">
<%out.println("快来学习如何开发 JSP 网站吧！");
%>
</p>
<%jsp:include page="bottom.txt" %>
```

尽管运行结果没区别，但是 include 动作指令和 include 指令还是有本质上的区别。include 动作指令在 JSP 页面被请求时才将指定文件的内容插入到服务器的相应输出中，而 include 指令在 JSP 文件被编译时就会插入文本或代码。

（2）forward 动作指令

forward 动作指令用来重定向网页，即从一个页面转向另一个页面。

forward 动作指令的格式如下：

```
<jsp:forward page="文件路径">
```

例如：利用<jsp：forward page＝"inlcudeTest.jsp">可以从当前页面跳转到 inlcudeTest.jsp 页面。

forward 指令只有一个 page 属性。如果想向目标文件传递参数，需要使用 param 动作指令。

（3）param 动作指令

param 动作指令通常与 include、forward 等其他动作指令配套使用，作用是传送参数。

param 动作指令的格式如下：

```
<jsp: param name="参数名"value="参数的值"/>
```

【例 6-16】　param 动作指令示例

① 新建文件 helloParam.jsp，代码如程序 6-16 所示。

【程序 6-16】　helloParam.jsp 文件代码

Example Source Code：

```
<%@ page contentType="text/html;charset=gb2312" %>
<html>
<head>
<title>param 指令示例</title>
</head>
<body>
<%
out.println("接收的参数 p 的值为："+request.getParameter("p"));
%>
```



```
</body>
</html>
```

② 新建文件 paramTest.jsp，代码如程序 6-17 所示。

【程序 6-17】 paramTest.jsp 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312" %>
<html>
<body>
<%
String p=null;
p="Hello ";
%>
<jsp:forward page="helloParam.jsp">
<jsp:param name="p" value="<%=p %>" />
</jsp:forward>
</body>
</html>
```

paramTest.jsp 文件在 forward 动作指令中指向了 helloParam.jsp 页面，并传递参数，<%=p %>是 JSP 表达式，得到变量 p 的值。helloParam.jsp 页面接收参数 p 的值，运行结果如图 6-7 所示。

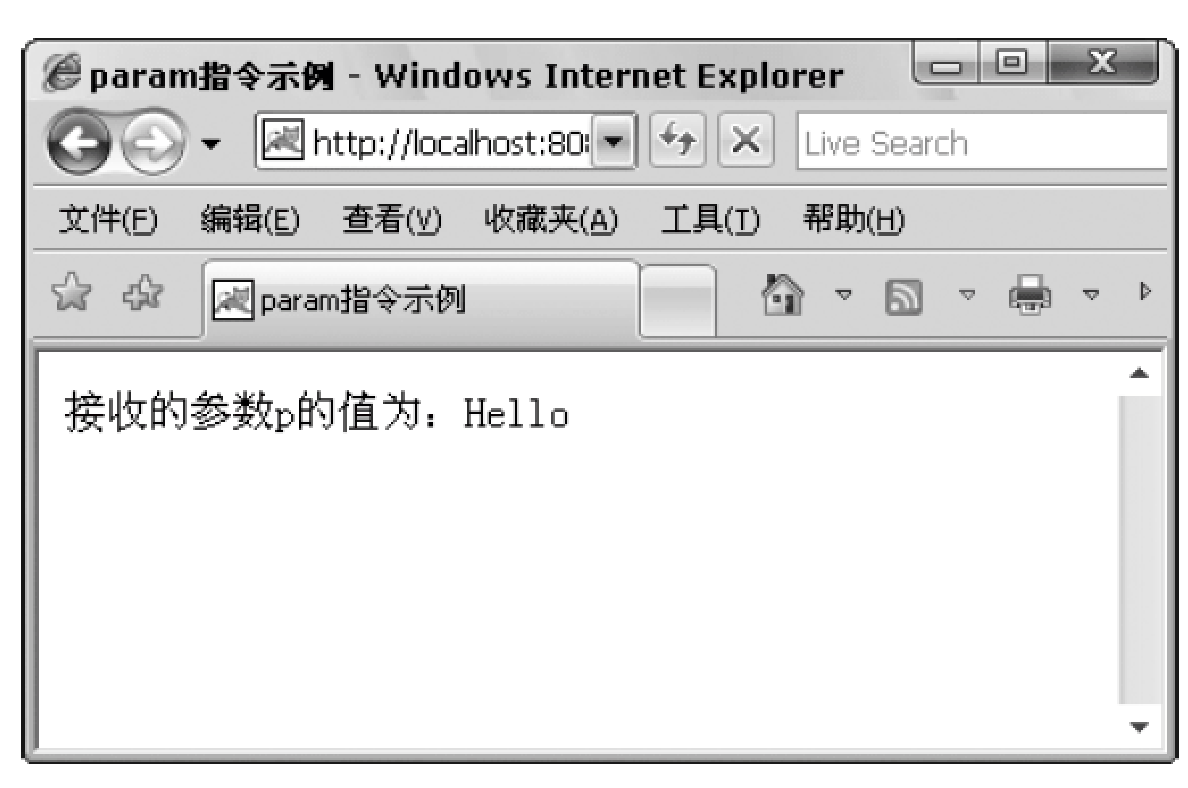


图 6-7　param 指令示例的运行结果

3. JSP 脚本元素

JSP 脚本元素用来插入 Java 代码，这些代码将出现在由当前 JSP 页面生成的 Servlet 中。脚本元素有三种：

（1）声明变量或方法

声明是指在 JSP 程序中定义合法的变量和方法。

声明格式：

```
<%! code %>
```

可以声明变量，声明的变量相当于类的成员变量。

举例：



```
<%!
  String name;
  %>
```

可以声明方法,声明的方法可以为静态方法或非静态方法,有关键字 static 修饰的方法是静态方法。静态方法的作用域是类作用域,而非静态方法的作用域是对象作用域,即只能通过类的一个对象访问非静态方法。

【例 6-17】 声明方法示例,如程序 6-18 所示。

【程序 6-18】 声明方法示例代码

Example Source Code:

```
<%!
public static String func1(){
    return("在静态方法 func1()中");
}
public String func2(){
    return("在非静态方法 func2()中");
}
%>
```

上面代码声明了两个方法:静态方法 fun1( )和非静态方法 fun2( )。

(2) 表达式

JSP 表达式用来把 Java 数据直接插入到输出中。

JSP 表达式的格式如下:

```
<%=表达式%>
```

计算表达式后得到的结果会自动转换为字符串,然后插入在这个表达式在 JSP 文件中的位置,显示在网页上。

【例 6-18】 JSP 表达式示例,如程序 6-19 所示。

【程序 6-19】 JSP 表达式示例代码

Example Source Code:

```
<%int x=2;%>
<%=5*x%>
```

上面代码段在网页上输出的结果是 10。

(3) Java 脚本代码

Java 脚本代码(Java Scriptlet)是指在 JSP 页面中的"<%"和"%>"之间插入的 Java 程序代码,其语法格式为:

```
<%Java脚本代码%>
```

【例 6-19】 Java 脚本代码示例,如程序 6-20 所示。

【程序 6-20】 Java 脚本代码示例代码

Example Source Code:

```
<%
for(int i=0; i<10; i++)
```



```
out.println(i+"<br>");
%>
```

这段代码的功能是输出 10 行数字，每行的数字从上到下依次为 0～9。

6.3.1 任务描述

根据条件判断，实现不同 JSP 页面的跳转。

6.3.2 任务实现

需要写 3 个 JSP 文件，分 3 个任务来完成。

任务 1　编写简单的登录成功页面 login-success.jsp

任务 2　编写登录失败页面 login-failture.jsp

任务 3　修改 6.2 节中的文件 admin_check.jsp

任务 1：编写简单的登录成功页面 login_success.jsp

代码如程序 6-21 所示。

【程序 6-21】　login_success.jsp 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312" %>
<html>
<head>
<title>登录成功页面</title>
</head>
<body>
<%
out.println("用户登录成功");
%>
</body>
</html>
```

任务 2：编写登录失败页面 login_failture.jsp

代码如程序 6-22 所示。

【程序 6-22】　login_failture.jsp 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312" %>
<html>
<head>
<title>登录失败页面</title>
</head>
<body>
<%
out.println("用户登录失败");
%>
```



```
</body>
</html>
```

任务 3：修改 6.2 节中的文件 admin_check.jsp

在 admin_check.jsp 中，实现的功能是：如果信息匹配，跳转到登录成功页面，如果信息不匹配，跳转到登录失败页面，代码如程序 6-23 所示。

【程序 6-23】 admin_check.jsp 文件代码

Example Source Code：

```
<%@page contentType ="text/html;charset=gb2312" %>
<html>
<head>
<title>登录验证页面</title>
</head>
<body>
<%String username="admin";
    String password;
    password="000000";
  if( username.equals("admin") && password.equals("000000"))
  {
%>
<jsp:forward page="login_success.jsp" />              //跳转到登录成功页面
<%
  }
else
{
%>
<jsp:forward page="login_failure.jsp" />;             //跳转到登录失败页面
<%
}
%>
</body>
</html>
```

程序的运行结果如图 6-8 所示。



图 6-8　页面跳转结果



因为给两个变量 name 和 password 所赋的值与判断条件中的值一致，所以由登录验证页面跳转到登录成功页面。

6.3.3 相关知识拓展

1. JavaBean 的含义

JavaBean 是 Java 程序设计中的一种组件技术。简单说，JavaBean 就是一个类，是封装了属性和方法，并用来完成某种特定功能的类。

2. 编写 JavaBean

编写 JavaBean 就是编写一个 Java 类，设计这个类的属性和方法。如果成员变量名为 xxxx，则相应有两个用来得到成员变量值和设置成员变量值的方法，分别是 getXxxx( )和 setXxxx( )。如果成员变量是 boolean 类型，那么还应该有一个 isXxxx( )方法。访问成员变量的方法都应该用 public 修饰，即公有方法。

3. userBean 动作指令

userBean 动作指令是一个重要的动作指令，用来在 JSP 中创建并使用一个 JavaBean。

在 JSP 中声明一个 JavaBean 的方法如下：

```
<jsp:useBean id="bean的名称" class="包名.类名"scope="有效范围">
</jsp:useBean>
```

其中，id 属性指定 JavaBean 实例名；class 指定类的引用；scope 指定有效范围，可以取 page、request、session、application 四个值。page 表示 JavaBean 的有效范围为当前页面，request 表示 JavaBean 可以被同一用户请求的页面访问，session 表示 JavaBean 在当前 http 会话的生命周期内可被所有页面访问，application 表示 JavaBean 的有效范围为第一次访问直到 Web 服务器关闭。

通过<jsp:useBean>动作指令在 JSP 页面中声明了 JavaBean 类实例之后，就可以使用<jsp:setProperty >和<jsp:getProperty >指令设置或读取 JavaBean 的属性。当然，也可以使用 JSP 脚本程序或表达式直接调用 JavaBean 对象的公有方法。

【例 6-20】 JavaBean 示例

(1) 新建一个 Java 文件，文件名为 Table.java，代码如程序 6-24 所示。

【程序 6-24】 Table.jsp 文件代码

Example Source Code:

```
package packTable;
public class Table{
    double length;
    double width;
    double height;
    public Table( ){
        length=0;
```



```
            width=0;
            height=0;
    }
    public double getHeight( ) {
            return height;
        }
        public void setHeight(double height) {
            this.height=height;
        }
        public double getLength( ) {
            return length;
        }
        public void setLength(double length) {
            this.length=length;
        }
        public double getWidth( ) {
            return width;
        }
        public void setWidth(double width) {
            this.width=width;
        }
}
```

（2）部署 JavaBean。

编译 Table. java，生成 Table. class，Table 类位于包 packTable 中，将该包复制到“WEB-INF\classes”目录下，这称为在 Tomcat 中部署 JavaBean。

（3）编写 JSP 文件，文件名为 javabeanJSP. jsp，代码如程序 6-25 所示。

【程序 6-25】 javabeanJSP. jsp 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312" %>
<%@page import="packTable.Table" %>          //导入 JavaBean 类，即 packTable 包中的 Table 类
<html>
<head>
<title>JavaBean 示例</title>
</head>
<body>
<jsp:useBean id="table" scope="page" class="packTable.Table">          //声明 JavaBean 类的实例
</jsp:useBean>
<%
table.setLength(20);                                  //调用 setXxxx( )方法，设置属性值
table.setWidth(10);
table.setHeight(30);
out.println("桌子长度为："+box.getLength( )+"<br>");
                                                      //调用 getXxxx( )方法，获取属性值
out.println("桌子宽度为："+box.getWidth( )+"<br>");
out.println("桌子高度为："+box.getHeight( )+"<br>");
%>
</body>
</html>
```



该程序中首先用 import 语句导入了 packTable. Table 类，然后生成 JevaBean 的实例 table，用 setXxxx( )设定 table 的长、宽、高，用 getXxxx( )方法得到 table 的长、宽、高并输出。

程序的运行结果如图 6-9 所示。

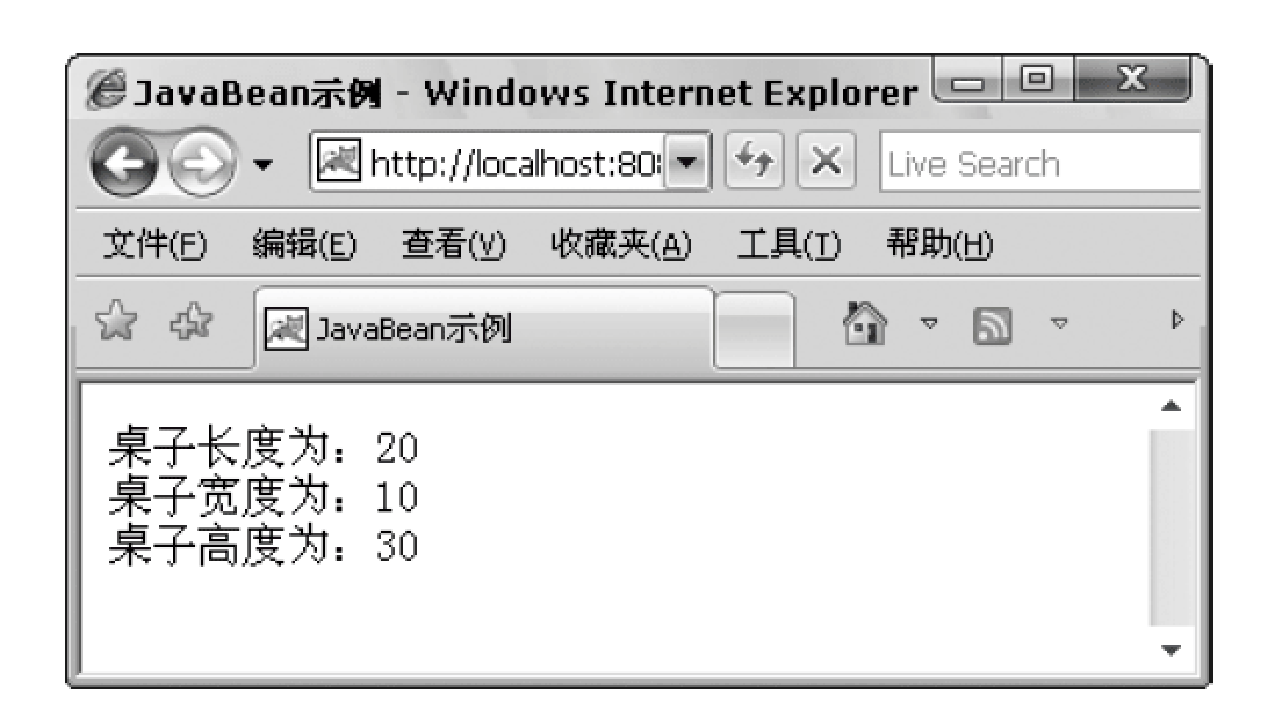


图 6-9　在 JSP 中使用 JavaBean

6.3.4　经验总结

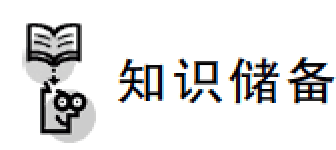
在编写 JSP 程序时，经常会出现中文字符显示出来是乱码的问题。碰到该类问题，需要考虑中文字符的处理。

一个网站的主导航条一般是统一的，所以经常把主导航条写成一个单独的网页，通过包含指令在其他 JSP 页面中实现导航条，这样可以减少编码的工作量，避免重复编码。

在 JSP 程序中，经常要进行页面的重定向，一般是在页面程序中加入判断语句，当满足条件时转向某一个页面，不满足条件时则转向另一个页面。

掌握 JavaBean 的开发和使用，有利于提高 JSP 开发的效率和程序代码的可重用性。通过 JavaBean，可以较大程度地实现 JSP 页面中静态内容和动态内容的分离。

6.4　JSP 内置对象



为了便于 Web 应用程序的开发，JSP 页面中内置了一些默认的对象，即 JSP 内置对象。内置对象在 JSP 页面中不用创建，可以直接使用。JSP 内置对象一共有 9 个：request、response、out、session、application、config、exception、page、pageContext。这些对象在服务器端和客户端交互的过程中分别完成不同的功能。为完成本节任务，需要使用下面三个最常用的内置对象：request、response 和 out。

1. request 对象

request 对象与 HTTPServletRequest 类关联，是 javax. servlet. ServletRequest 的一个子类。request 对象封装了客户端提交的数据信息，包括访问者输入的信息以及客户端环境的一些信息。客户端可通过 HTML 表单或者在网页地址后面使用参数的方法提交数据，再用 request 对象的相关方法来获取提交的各种数据。



request 对象的一个常用方法是 getParameter( )，此方法可以获得指定的客户端提交的参数值。方法调用格式如下：

```
request.getParameter(String str)
```

其中，参数 str 是指定的参数名称。

【例 6-21】 request 对象示例

```
request.getParameter("uname");        //uname 代表用户名文本框
```

在表单的提交请求处理页面中，可以通过上面的代码获取名称为 uname 的表单项的值。

2. response 对象

response 对象与 HTTPServletResponse 类关联，该对象的主要作用是响应客户端请求，将 JSP 处理数据后的结果传回给客户端。客户端浏览器每访问一次 Web 服务器页面时都会向服务器提交一次请求，服务器在收到来自客户端浏览器发来的请求后响应请求。JSP 通过 request 对象控制客户浏览器的请求，通过 response 对象对客户浏览器进行响应，二者的作用正好相反。

response 对象的一个常用方法是 sendRedirect( )，此方法可以将页面重定向至其他页面，重定向后在浏览器地址栏上会出现重定向页面的 URL。与前面介绍过的 forward 动作的区别是：forward 动作实现重定向后地址栏显示的 URL 未变，而 response 对象实现重定向后地址栏显示的所定向页面的 URL。

方法调用格式如下：

response.sendRedirect(String url)，其中，参数 url 可以是合法的 url 地址，也可以是文件路径。

【例 6-22】 request 对象示例

```
response.sendRedirect("loginSuccess.jsp");
```

在某一页面中，可以利用该语句跳转到 loginSuccess.jsp 页面。

3. out 对象

out 对象用来把结果输出到网页上。在前面的例子中已经多次使用过，其最为常用的方法是 print( )和 println( )方法。利用 out.print( )和 out.println( )都可以输出任意类型的数据。

print( )和 println( )的区别是：out.println( )在输出的数据后面加换行符，而 out.print( )不会在输出的数据后加换行符。但是在 IE 浏览器中显示 HTML 网页时，这种区别并不存在，因为 println( )方法所加的换行符在浏览器中并不起作用。如果用记事本查看 HTML 源代码时可以看到这个差别。

6.4.1　任务描述

编写 JSP 登录验证程序。

6.4.2　任务实现

需要 4 个 JSP 文件：login.jsp、login_check.jsp、login_sucess.jsp、login_failure.jsp，分



4个任务完成。

任务1　编写登录页面

任务2　改写 login-cheek. jsp，获取表单提交的值，进行登录验证和页面跳转

任务3　改写登录成功页面 login-sucess. jsp，显示获取的用户名

任务4　改写登录失败页面 login-failture. jsp，显示用户名或密码不正确

任务1：编写登录页面

新建文件 login. jsp，在<body>与</body>之间添加登录表单的代码。表单中包括用户名和密码框、登录和重置按钮，表单信息提交给 login_check. jsp 处理。

创建表单在前面的 HTML 中介绍过，此处不再重复，表单关键代码如程序6-26所示。

【程序6-26】　login. jsp 文件代码

Example Source Code：

```
<form action="login_check.jsp" method="post">
    <table>
        <tr>
            <td>用户名：</td>
            <td><input type="text" name="uname"></td>
        </tr>
        <tr>
            <td>密   码：</td>
            <td><input type="password" name="upassword"></td>
        </tr>
        <tr>
            <td colspan="2">
            <input type="submit" value="登录">
            <input type="reset" value="重置">
            </td>
        </tr>
    </table>
</form>
```

登录页面的运行结果如图6-10所示，用户可以在文本框中输入用户名和密码，单击【登录】按钮进行登录验证。



图6-10　登录页面



任务 2：改写 login_check.jsp，获取表单提交的值，进行登录验证和页面跳转

在登录验证页面中，利用 request 对象获取用户输入的用户名和密码，利用 response 对象进行页面重定向，如果用户名和密码正确，则跳转到 login_sucess.jsp 页面，否则跳转到 login_failure.jsp 页面。

其中，实现判断以及页面跳转的关键代码如程序 6-27 所示。

【程序 6-27】　login_check.jsp 文件代码

Example Source Code：

```
<%
    //接收请求的内容
    String name=request.getParameter("uname");
    String password=request.getParameter("upassword");
%>
<%
    //判断用户名及密码
    if("admin".equals(name)&&"zhy".equals(password))
    {
        //合法用户
%>
        <jsp:forward page="login_success.jsp"/>           //跳转到登录成功页面
<%
    }
    else
    {
        //非法用户
%>
        <jsp:forward page="login_failture.jsp" />          //跳转到登录失败页面
<%
    }
%>
```

任务 3：改写登录成功页面 login_sucess.jsp，显示获取的用户名

代码如程序 6-28 所示。

【程序 6-28】　login_sucess.jsp 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312"%>
<html>
<head>
    <title>登录成功页面</title>
</head>
<body>
<center>
    <h2>登录成功</h2>
    <h3>欢迎<font color="red" size="15">
        <%=request.getParameter("uname")%>                  //输出获取的用户名
    </font>光临!!!</h3>
</center>
</body>
</html>
```



任务 4：改写登录失败页面 login_failture.jsp，显示用户名或密码不正确

代码如程序 6-29 所示。

【程序 6-29】 login_failture.jsp 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312" %>
<html>
<head>
<title>登录失败页面</title>
</head>
<body>
<h2 align="center">登录失败</h2>
<%
    out.println("用户名或密码不正确！");                    //输出用户名或密码不正确
%>
</body>
</html>
```

程序运行过程：

(1) 首先进入登录页面，在登录页面中输入用户名和密码，如图 6-11 所示。



图 6-11 输入用户信息

(2) 如果输入用户名 admin 和密码 zhy，则单击【登录】按钮后进入登录成功页面，如图 6-12所示。

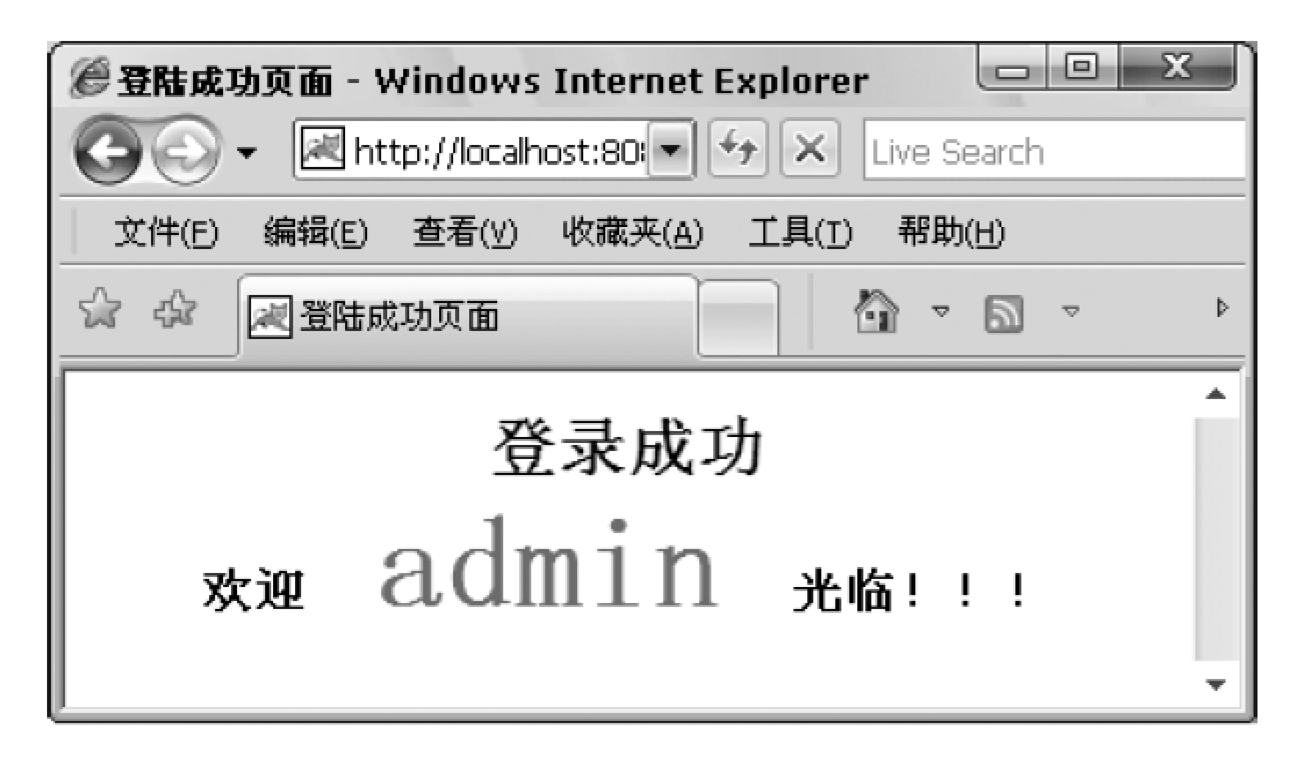


图 6-12 登录成功页面



(3) 如果在登录页面上输入不正确的用户名或密码，则进入登录失败页面，如图 6-13 所示。



图 6-13　登录失败页面

6.4.3　相关知识拓展

session 对象和 application 对象也是常用的内置对象。

1. session 对象

session 对象用来保存网站与每个访问者会话期间需要保持的数据信息，这样就方便在会话期间处理程序。所以 session 对象的作用是在多个页面共享某一信息。例如，可以用 session 变量记住用户名，以后不必在其他的网页中重复输入。当用户关闭某个 Web 服务器上的所有网页时，此服务器与客户机的 session 对象及其变量会自动消失。

session 对象的常用方法有 setAttribute( )和 getAttribute( )。

setAttribute( )方法用来设置指定 session 对象中数据对象的值，其调用格式如下：

```
session.setAttribute(String name, Object value)
```

其中，第一个参数 name 指出要设置的 session 对象中数据对象的名称；第二个参数 value 是这个对象对应的值。如果指定的对象不存在，则新加入一个；如果已存在，则该值会被更改。

getAttribute( )方法用来获得指定的 session 数据对象，其调用格式如下：

```
session.getAttribute(String name)
```

其中，参数 name 是指定 session 对象的名称。该方法返回一个 Object 对象，任何对象都可以加入到 session 中，因此对返回的对象要用强制类型转换的方法，将得到的对象转换为此对象原来的类型。

【例 6-23】　session 对象示例

新建文件 sessionTest.jsp，代码如程序 6-30 所示。

【程序 6-30】　sessionTest.jsp 文件代码

Example Source Code:

```
<%@page contentType="text/html;charset=gb2312" %>
```



```
<html>
<head>
<title>存取 session 数据</title>
</head>
<body>
<%
    int num=0;
    Object obj=session.getAttribute("num");    //获取 session 中名称为 num 的数据对象
    if(obj==null){
        session.setAttribute("num",String.valueOf(num));
                                               //在 session 中建立 num 数据对象
    }
    else
    {
        num=Integer.parseInt(obj.toString());
        num+=1;
        session.setAttribute("num",String.valueOf(num));
                                               //修改 session 中 num 数据对象的值
    }
%>
session 中的 num 值为:
<%=num%>
</body>
</html>
```

本例模拟实现网站的用户在线统计,第一次运行的页面结果如图 6-14 所示。

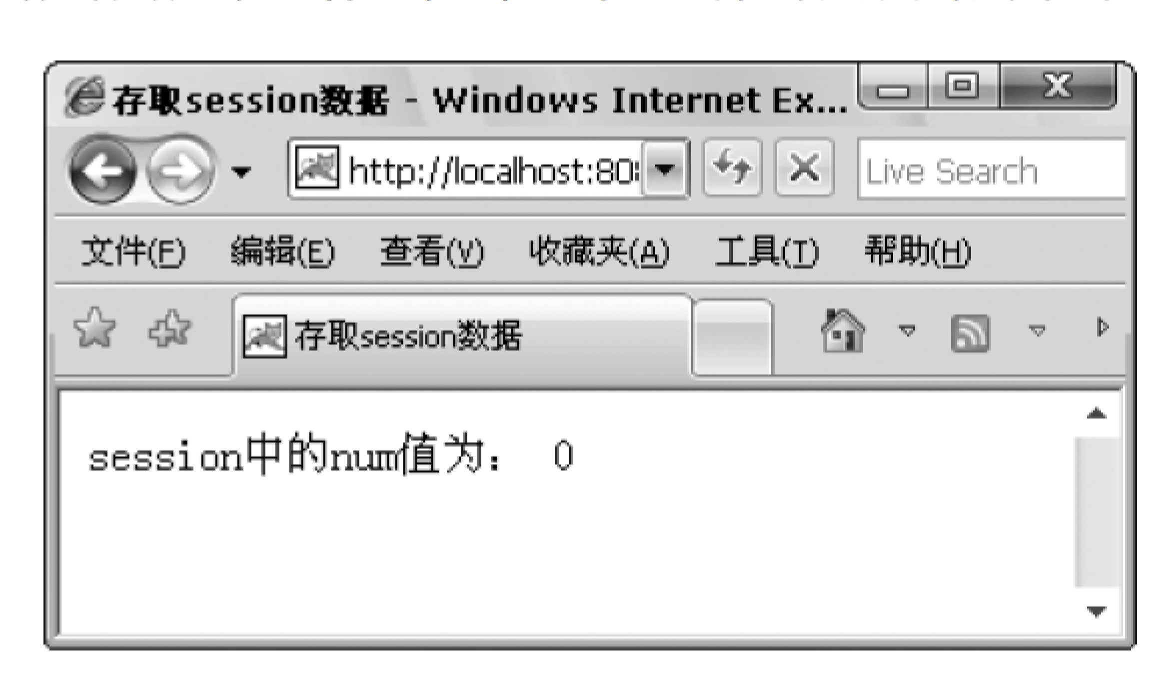


图 6-14 session 示例运行结果 1

刷新一次浏览器,session 中的变量值会加 1,页面结果如图 6-15 所示。



图 6-15 session 示例运行结果 2



2．application 对象

application 对象用来在多个程序或者多个用户之间共享数据，用户使用的所有 application 对象都是一样的，这与 session 对象不同。服务器一旦启动，就会自动创建 application 对象，并一直保持。当服务器关闭时，application 对象会自动消失。

application 对象也有 setAttribute( )和 getAttribute( )方法。

application 对象的 setAttribute( )方法用来设置指定名称的 application 数据对象的属性值，其调用格式为：

```
application.setAttribute(String name,Object object)
```

其中，第一个参数 name 指出要设置的 application 对象中数据对象的名称；第二个参数 object 是这个对象对应的值，其类型是 Object，即可以是任意类型。如果 application 中不存在对象，则添加该对象；如果已存在，则将其覆盖。

application 的 getAttribute( )方法得到指定的 application 中数据对象的值。其调用格式如下：

```
application.getAttribute(String name)
```

其中，参数 name 是指定 application 中数据对象的名称。

例 6-23 中用 session 对象统计访问人数，每刷新一次，计数器加 1，因此计数不准确。计数应当根据是否是一个新的会话来判断这次访问网站的人是否是一个新访问者，可以利用 application 来保存访问网站的人数。

【例 6-24】 application 对象示例

新建文件 applicationTest.jsp，利用 application 对象实现计数器程序，代码如程序 6-31 所示。

【程序 6-31】 applicationTest.jsp 文件代码

Example Source Code：

```
<%@page contentType="text/html;charset=gb2312" %>
<html>
<head>
<title>网站计数器</title>
</head>
<body>
<%!
    synchronized void countPeople()                          //串行化计数函数
{
        ServletContext application=getServletContext();      //得到 application 上下文
        Integer num=(Integer)application.getAttribute("Count");
        if(num==null)                                         //如果是第一个访问者
{
            num=new Integer(1);
            application.setAttribute("Count",num);
        }
        else
        {
```



```
num=new Integer(num.intValue()+1);
            application.setAttribute("Count",num);
        }
    }
%>
<%if(session.isNew())                          //如果是一个新的会话
countPeople();
Integer Number=(Integer)application.getAttribute("Count");
%>
您好,您是本站的第<%=Number %>位客人!
</body>
</html>
```

程序运行结果如图 6-16 所示。当刷新页面时,其数值并不加 1,只有关闭了所有窗口再重新访问时,才会加 1。

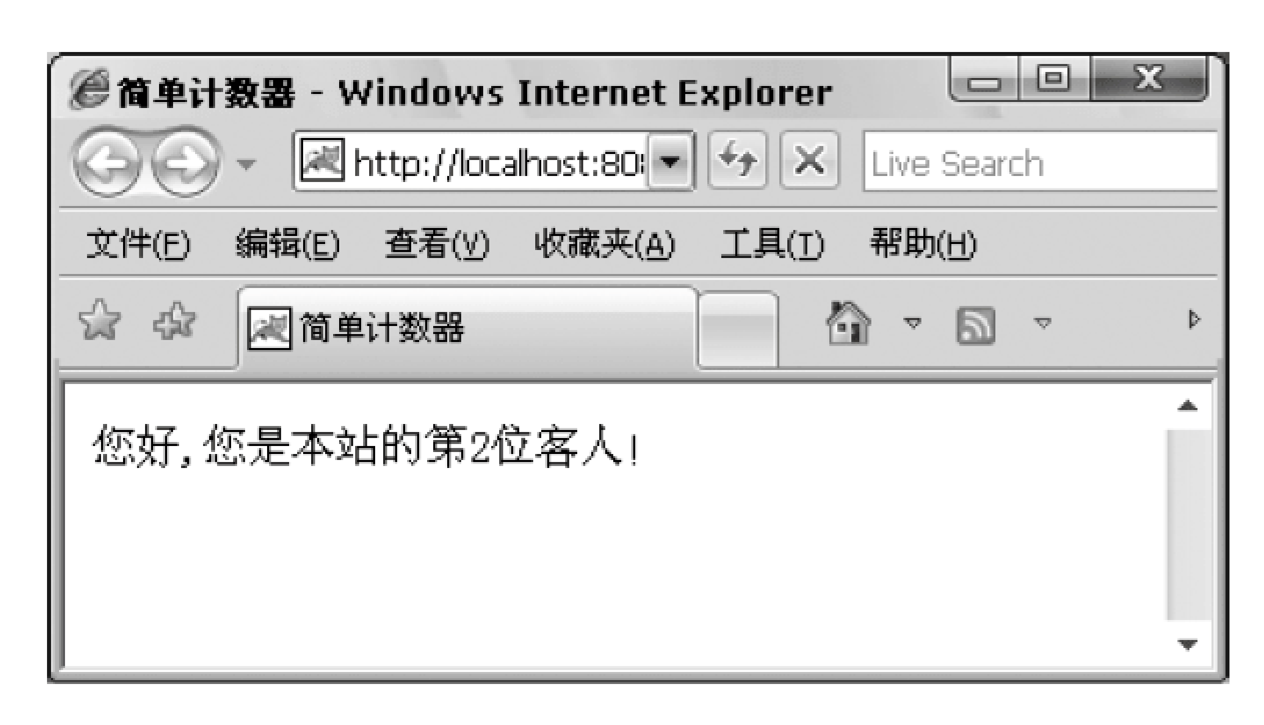


图 6-16　网站计数器

6.4.4　经验总结

request、response、out、session 和 application 这 5 个内置对象在 JSP 编程中使用非常频繁,是要掌握的重点内容,一定要通过实例体会它们的用法。至于其他内置对象,编程中很少用到,因此可以暂且不学。

在 JSP 编程中,服务器程序与客户端交互的最常用的方法是通过表单提交数据,利用 request 对象可以得到表单中的相应值。request 对象的对应对象是 response,response 对象用于响应客户端请求,向客户端发送数据。

HTTP 是一个无状态的协议,不保留会话间的数据,可通过 session 对象保留会话期间需要的数据信息。application 对象用于提供一些全局的数据和对象,保留在服务器端。

要使用 session 对象,浏览器必须支持 cookie。如果客户端浏览器禁止 cookie,则无法从客户端浏览器中取得作为 cookie 的 session ID,即 session 无效。

6.5　实　　训

实现“新闻类别网站”的登录验证功能。



第7章　实现网站服务器端的功能

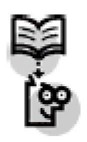


网站服务器端的功能，就是整个系统的业务层，是负责对数据进行处理的，即获取并执行用户操作、对数据进行存取和形成并响应用户操作等。

本系统网站服务器端功能的实现主要由 Java 语言实现，在整个开发过程中涉及较多的 Java 语言相关知识。

本章遵循软件工程的原理，在前几章进行相关设计的基础上，从编写公共模块文件、主要功能模块、其他功能模块和网站安全性设计几个方面对系统的整个开发过程进行了详细而全面的讲解。

7.1　编写公共模块文件



知识储备

1. 包的作用

(1) 包用于组织 Java 文件，把 Java 文件分类到不同的文件夹，便于查找和分类。

(2) 包可以减少文件命名重复的概率，定位一个 Java 文件不仅仅是它的名字，还要加上包名，无形中加长了文件名字。

2. 建立包结构的四个指导性原则

(1) 包内高内聚，但在某些分布式环境下可能会打破这种原则。

(2) 隐藏细节，特别是实现的细节。

(3) 面向问题域（即面向业务），使用该领域的一般概念组织包结构，而不是使概念支离破碎。

(4) 一个包内的类可以被共同重用。

3. 建立包间关系的四条指导性原则

(1) 包间低耦合。

(2) 无循环依赖（ADP）。

(3) 稳定依赖（SDP），尽量依赖稳定的包（意味着难以改变）。

(4) 抽象依赖（SAP），稳定的包要用抽象的层（or 接口）隔离。

4. Java 的导入（import）

Java 导入类的两种方式是：单类型导入（single－type－import，例如 import java. io.



File)，按需类型导入(type—import—on—demand，例如 import java.io.*)。

(1) 单类型导入

对于单类型导入，仅仅导入一个 public 类或者接口。

单类型导入类时，Java 编译器会从启动目录(bootstrap)、扩展目录(extension)和用户类路径依次去定位需要导入的类，而这些目录仅仅是给出了类的顶层目录。Java 编译器的类文件定位方法可以按如下公式来理解：

顶层路径名\包名\文件名.class=绝对路径

可以看出，采用单类型导入类时，因为包名和文件名都已经确定，所以可以一次性查找定位。使用单类型导入至少有以下两点好处：

① 提高编译速度。

② 避免命名冲突。

但是使用单类型导入会使用 import 语句，看起来较长。

(2) 按需类型导入

按需类型导入，并不是导入一个包下的所有类，而是按需导入，即并非导入整个包，而仅仅导入当前需要使用的类。

按需类型导入类时，Java 编译器会把包名和文件名进行排列组合，然后对类文件所有可能存在的路径进行查找定位。例如：

```
package com;
import java.io.*;
import java.util.*;
```

当类文件中用到了 File 类，那么可能出现 File 类的地方有：

① File：File 类属于无名包，就是说 File 类没有 package 语句，编译器会首先搜索无名包。

② com.File：File 类属于当前包。

③ java.lang.File：编译器会自动导入 java.lang 包。

④ java.io.File。

⑤ java.util.File。

Java 编译器找到 java.io.File 类之后并不会停止下一步的寻找，而要把所有的可能性都查找完以确定是否有类导入冲突。假设此时的顶层路径有三个，那么编译器就会进行 3×5=15次查找。

所以，按需类型导入类并不会降低 Java 代码的执行效率，但会影响到 Java 代码的编译速度。

7.1.1 任务描述

利用 Eclipse 编写“新闻类别网站”的公共模块文件。



7.1.2　任务实现

在每个系统中，都需要对字符串、页面参数、网页 HTML 代码、Cookie 参数进行处理来实现系统的交互，可以说对系统的实现都是在进行字符串的格式化处理。因此，根据网页整体设计的功能抽象，需要对各种类型的字符串进行特殊的输入输出处理，在此根据实际的需求和经验，总结了“新闻类别网站”的功能组件设计，共分五个任务来实现。

在实际的操作中，我们将建立对应的包结构，用来存放相关的文件，在本项目中存放公用类的包结构为：com\news\util。

任务 1　构建包结构
任务 2　编写页面参数处理模块
任务 3　编写格式化字符串模块
任务 4　编写表现层页面公共模块
任务 5　编写表现层的 JSP 页面框架结构

任务 1：构建包结构

具体操作步骤如下：

(1) 在 Eclipse 界面左侧的【项目资源管理器】中“右击”并选择新建的 Web 项目 New，然后选择【新建】|【包】命令，如图 7-1 所示。

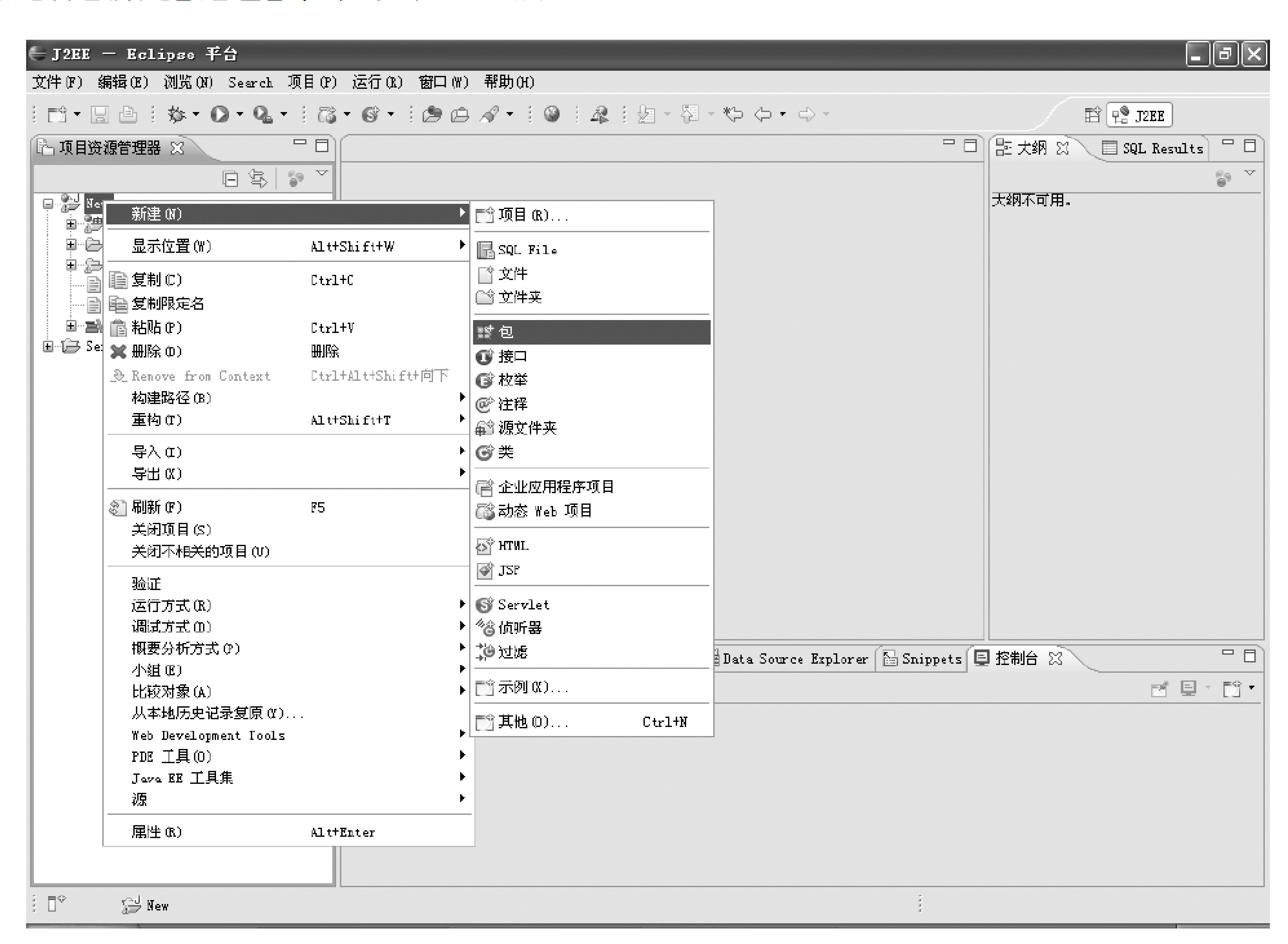


图 7-1　选择【新建】|【包】命令

(2) 在弹出的【新建 Java 包】对话框的【名称】文本框中输入“util”，单击【完成】按钮，如图 7-2 所示。



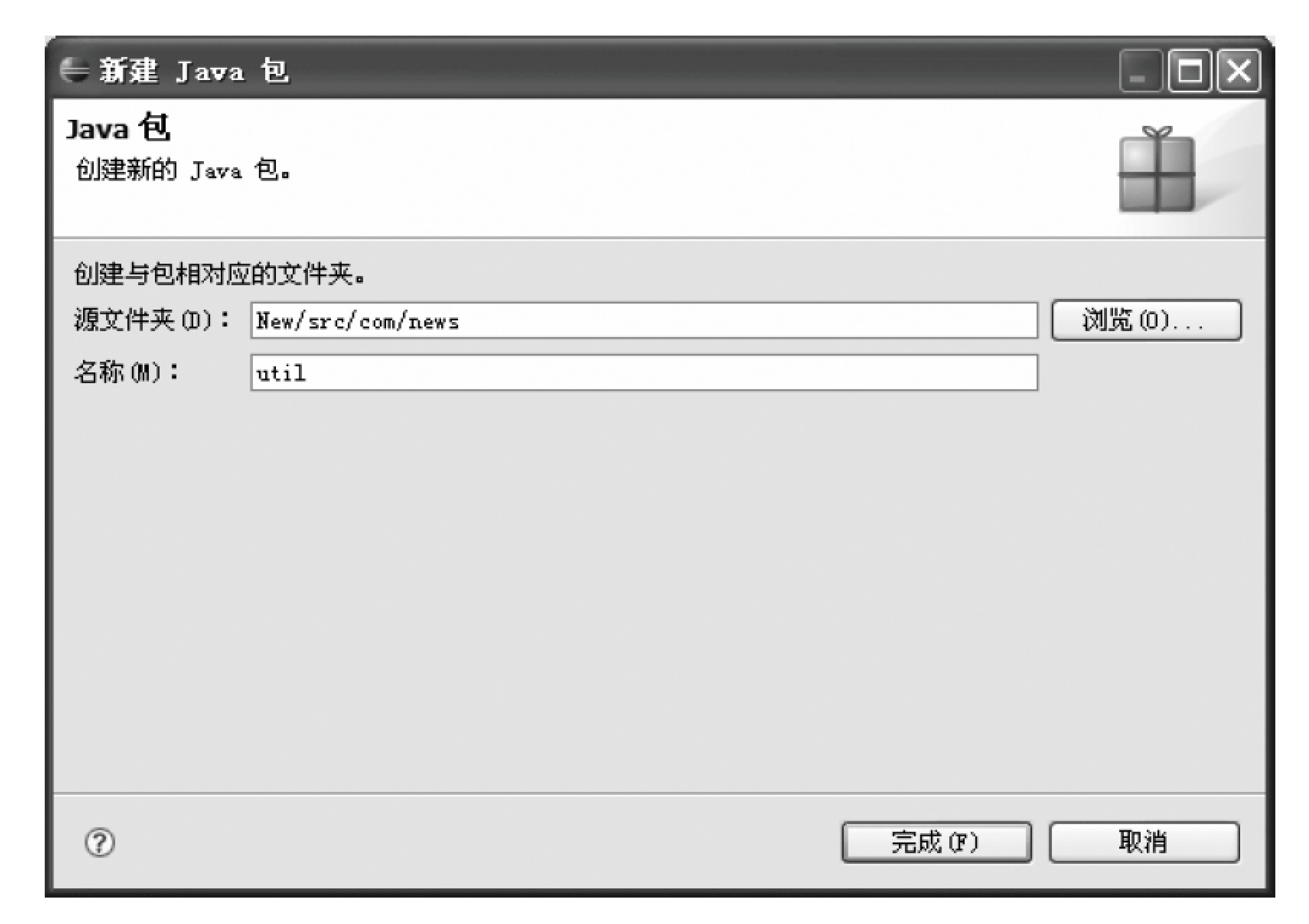


图 7-2 【新建 Java 包】对话框

任务 2：编写页面参数处理模块

下面的类的代码把表单参数进行初始化并取出值来，用法为 ParamUtil. getString（“txt_name”），因为本例声明 getString 为静态函数。此函数有同名重载的函数，当参数取出的值为空时，一个返回 null，一个返回默认值。具体操作步骤如下：

（1）在 Eclipse 中新建页面参数处理模块 ParamUtil 类文件，保存到任务 1 所建的包中，文件名为 ParamUtil. java。

（2）在 ParamUtil 类中添加如程序 7-1 所示的代码。

【程序 7-1】 getString(request，paramName)方法

Example Source Code：

```
public static String getString(HttpServletRequest request,String paramName){
    String temp=request.getParameter(paramName);
    if(temp!=null&&!temp.equals(""))
        return temp;
    else
        return null;
}
public static String getString(HttpServletRequest request, String paramName, String
defaultString){
    String temp=getString(request,paramName);
    if(temp==null)
        temp=defaultString;
    return temp;
}
public static int getInt(HttpServletRequest request, String paramName) throws
NumberFormatException{
        return Integer.parseInt(getString(request,paramName));
}
```



```
public static int getInt(HttpServletRequest request,String paramName,int defaultInt){
    try{
        String temp=getString(request,paramName);
        if(temp==null)
            return defaultInt;
        else
            return Integer.parseInt(temp);
    }
    catch(NumberFormatException e){
        e.printStackTrace();
        return 0;
    }
}
```

任务 3：编写格式化字符串模块

该类为对一些不规则的数据进行格式化处理的类，如对日期时间的不同格式进行格式化，以便达到系统要求的精确度，另外此类还补充了字符处理类所没有的一些功能，如字符串替换，将源字符串中的旧字符串全部换成新的字符串，将字符串格式化成 HTML 代码输出，或将普通字符串格式化成数据库认可的字符串格式等。

Format.java 包括的主要功能方法有：

public static String getDateTime( )

public static String getStrDate(String DateString, int ShowType)

public static boolean compareTo(String last, String now)

public static String toHtmlInput(String str)

public static String toHtml(String str)

public static String toSql(String str)

具体操作步骤如下：

(1) 在 Eclipse 中新建格式化字符串模块 Format 类文件，保存到任务 1 所建的包中，文件名为 Format.java。

(2) 在 Format 类中添加如程序 7-2 和程序 7-3 所示的代码。

程序 7-2 和程序 7-3 是对日期时间数据进行格式化处理的函数，将其转化为形如“2004-10-25 21:25:32”的格式，不同系统对时间要求的精确度可能不同，有的会以时间作为唯一关键的字段，所以要保证其唯一性，因此有必要进行处理。

【程序 7-2】 getDateTime( )方法

Example Source Code：

```
/**
 *获取系统时间
 *@return            格式化后的字符串日期
 */
public static String getDateTime() {
    SimpleDateFormat formatter=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    java.util.Date Now=new java.util.Date();
    String NDate=formatter.format(Now);
    return NDate;
```



```
    }
```

【程序 7-3】 getStrDate(String DateString, int ShowType)

Example Source Code:

```
/* *
 * 获取字符串日期,将指定的 DateString 按照 ShowType 指定的类型显示出来
 * @param DateString        字符串型日期
 * @param ShowType          日期显示的类型
 * @return                  变换后的字符串型日期
 * /
public static String getStrDate(String DateString, int ShowType) {
    try {
        SimpleDateFormat sdf=new SimpleDateFormat();
        Date now=sdf.parse(DateString);
        DateFormat date;
        switch (ShowType) {
        case 1: // "Y-m-d"
            date=DateFormat.getDateInstance(DateFormat.DAY_OF_YEAR_FIELD);
        case 2: // "Y-m-d H:I A":
            date=DateFormat.getDateInstance(DateFormat.AM_PM_FIELD);
        case 3:
            date=DateFormat.getDateInstance(DateFormat.FULL);
        case 4: // "YmdHIS":{
            date=DateFormat.getDateInstance(DateFormat.HOUR_OF_DAY1_FIELD);
        case 5: // "ym":
            date=DateFormat.getDateInstance(DateFormat.MONTH_FIELD);
        case 6: // "m-d":
            date=DateFormat.getDateInstance(DateFormat.MINUTE_FIELD);
        case 7: // "d":
            date=DateFormat.getDateInstance(DateFormat.FULL);
        default:
            date=DateFormat.getDateInstance(DateFormat.DEFAULT);
        }
        return date.format(now);
    } catch (ParseException e) {
        e.printStackTrace();
        return "d";
    }
}
```

(3) 在 Format 类中添加如程序 7-4 所示的代码。

程序 7-4 是对两个字符串进行比较的函数,为了有效地对两个字符串进行比较,有必要设置此方法进行相关的处理。

【程序 7-4】 compareTo(String last, String now)方法

Example Source Code:

```
/* *
 * 比较两个字符串替换,将 last 与 now 进行比较
```



```
 * @param last              老的字符串
 * @param now               新的字符串
 * @return                  两字符串是否相等
 * /
public static boolean compareTo(String last, String now) {
    try {
        DateFormat formatter=DateFormat.getDateInstance();
        Date temp1=formatter.parse(last);
        Date temp2=formatter.parse(now);
        if (temp1.after(temp2))
            return false;
        else if (temp1.before(temp2))
            return true;
    } catch (ParseException e) {
        e.printStackTrace();
    }
    return false;
}
```

(4) 在 Format 类中添加如程序 7-5 所示的代码。

程序 7-5 是将指定字符串中的指定旧字符替换为新字符，有时程序需要将一些指定的字符替换为新的字符，因此有必要进行处理。

【程序 7-5】 Replace(String source, String oldString, String newString)

Example Source Code:

```
/* *
 * 字符串替换，将 source 中的 oldString 全部换成 newString
 * @param source          源字符串
 * @param oldString       老的字符串
 * @param newString       新的字符串
 * @return                替换后的字符串
 * /
public static String Replace(String source, String oldString,String newString) {
    StringBuffer output=new StringBuffer();
    int lengthOfSource=source.length();              // 源字符串长度
    int lengthOfOld=oldString.length();              // 老字符串长度
    int posStart=0;                                  // 开始搜索位置
    int pos;                                         // 搜索到老字符串的位置
    while ((pos=source.indexOf(oldString, posStart))>=0) {
        output.append(source.substring(posStart, pos));
        output.append(newString);
        posStart=pos+ lengthOfOld;
    }
    if (posStart< lengthOfSource) {
        output.append(source.substring(posStart));
    }
    return output.toString();
}
```



(5) 在 Format 类中添加如程序 7-6 和程序 7-7 所示的代码。

程序 7-6 和程序 7-7 是将字符串格式化成 HTML 代码输出的函数,只转换特殊字符,适合于 HTML 中的表单区域,特殊符号如 &、<、>等在 HTML 中定义了相应的代码,为了更方便地提交给服务器进行解释,此函数提供了这种转换功能。

【程序 7-6】 toHtmlInput(String str)

Example Source Code:

```
/* *
 * 将字符串格式化成 HTML 代码输出,只转换特殊字符,适合于 HTML 中的表单区域
 * @param str          要格式化的字符串
 * @return             格式化后的字符串
 * /
public static String toHtmlInput(String str) {
    if (str==null)
        return null;
    String html=new String(str);
    html=Replace(html, "&", "&amp;");
    html=Replace(html, "<", "&lt;");
    html=Replace(html, ">", "&gt;");
    return html;
}
```

【程序 7-7】 toHtml(String str)

Example Source Code:

```
/* *
 * 将字符串格式化成 HTML 代码输出,除普通特殊字符外,还对空格、制表符和换行进行转换,以将内
容格式化输出,适合于 HTML 中的显示输出
 * @param str          要格式化的字符串
 * @return             格式化后的字符串
 * /
public static String toHtml(String str) {
    if (str==null)
        return null;
    String html=new String(str);
    html=toHtmlInput(html);
    html=Replace(html, "\r\n", "\n");
    html=Replace(html, "\n", "<br>\n");
    html=Replace(html, "\t", "     ");
    html=Replace(html, "  ", "  ");
    return html;
}
```

(6) 在 Format 类中添加如程序 7-8 所示的代码。

程序 7-8 也是将普通字符串格式化成数据库认可的字符串格式的函数,数据库提取出来的数据和程序中处理的数据,经常出现为初始化的现象,导致页面会显示 Null 等不正常的字符串,因此有必要进行处理。





【程序 7-8】　toSql(String str)方法

Example Source Code：

```
/* *
 * 将普通字符串格式化成数据库认可的字符串格式
 * @param str            要格式化的字符串
 * @return               合法的数据库字符串
 * /
public static String toSql(String str) {
    String sql=new String(str);
    return Replace(sql, "'", "''");
}
```

(7) 在 Format 类中添加如程序 7-9 所示的代码。

程序 7-9 是将普通字符串按照指定的数量显示出来，或者截掉多余的字符，在列表显示相关标题时，由于标题字符长度的不同，经常出现参差不齐的现象，导致页面显示不规范，因此有必要进行处理。

【程序 7-9】　CutStr(String str, int Len)方法

Example Source Code：

```
/* *
 * 将普通字符串 str 按照 Len 指定的长度显示，多余的字符将被截掉，并补以"…"。
 * @param str            要格式化的字符串
 * @param Len            字符串的长度
 * @return               合法的数据库字符串
 * /
public static String CutStr(String str, int Len) {
    if (str.length()>Len) {
        return str.substring(0, Len)+ "…";
    }
    return str;
}
```

任务 4：编写表现层页面公共模块

公共页面模块就是将网页头部、导航、版权和尾部等一些经常出现的部分进行封装，在需要这些部分的时候直接调用，尤其是在模块化设计盛行的今天，这种公共模块封装思想更是流行，而且这样的设计也有助于进行动态网站的维护，当需要改动站点时，只要在对应的模块中作修改就可以了，从而实现"一处修改，全部改变"的效果。

Config. java 包括的主要功能方法有：

```
public String PrintHead(String title)
public String PrintTop()
public String PrintFooter()
```

具体操作步骤如下：

(1) 在 Eclipse 中新建表现层页面公共模块 Config 类文件，保存到任务 1 所建的包中，文件名为 Config. java。

(2) 在 Config 类中添加如程序 7-10 所示的代码。



程序 7-10 是将网页的头部代码进行封装，为了更灵活的显示网页的标题，程序中设置了一个参数 title，这样就使页面的模块化处理更到位了。

【程序 7-10】 PrintHead(String title)方法

Example Source Code：

```
/*****************************************************************************************
 * 函数名：PrintHead
 * 作用：封装网页头部代码
 * 参数：title，网页标题
 * 返回值：字符型。返回网页头部代码
 * @throws Exception
*****************************************************************************************/
public String PrintHead(String title) throws Exception {
    try {
        StringBuffer sb=new StringBuffer();
         sb.append("<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Transitional//EN
         \" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd\">\n");
        sb.append("<html xmlns=\"http://www.w3.org/1999/xhtml\">\n");
        sb.append("<head>\n");
        sb.append("<meta http-equiv=\"Content-Type\" content=\"text/html; charset= gb2312
        \" />\n");
        sb.append("<title>"+title+"</title>\n");
        sb.append("<link href=\"Themes/default/Style.css\" rel=\"stylesheet\" type=\"
        text/css\" />\n");
        sb.append("<script language=\"javascript\" type=\"text/javascript\" src=\"
        Comment.js\"></script>\n");
        sb.append("</head>\n");
        sb.append("<body>\n");
        sb.append(PrintTop());
        return sb.toString();
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}
```

（3）在 Config 类中添加如程序 7-11 所示的代码。

程序 7-11 是将网页的导航代码进行封装，并利用循环实现了栏目信息的动态显示。

【程序 7-11】 PrintTop( )方法

Example Source Code：

```
/*****************************************************************************************
 * 函数名：PrintTop
 * 作用：封装网页导航代码
 * 参数：无
 * 返回值：字符型。返回网页导航代码
 * @throws Exception
*****************************************************************************************/
public String PrintTop() throws Exception {
```



```
    try {
        StringBuffer sb=new StringBuffer();
        sb.append("<div ID=\"Top\">\n");
        sb.append("  <div id=\"Logo\">\n");
        sb.append("<img src=\"Themes/default/Images/Flogo.jpg\" />\n");
        sb.append("</div>\n");
        sb.append("  <div ID=\"Banner\">\n");
        sb.append("<a href=\"http://211.82.32.36/\">学校首页</a>|\n");
        sb.append("<A href='#' name=home onclick=defaul_home();>设为首页</a>|\n");
        sb.append("<A href='#' onclick=defaul_addFavorite();>添加收藏</a>|\n");
        sb.append("<a href='mailto:guotianbing@sxftc.edu.cn'>联系我们</a>\n");
        sb.append("</div>\n");
        sb.append("  <div class=\"clear\"></div>\n");
        sb.append("</div>\n");
        sb.append("<div ID=\"MenuBar\">\n");
        sb.append("  <DIV ID=\"time\">\n");
        sb.append("<SCRIPT type=\"text/javascript\">Printtime()</SCRIPT>\n");
        sb.append("</DIV>\n");
        sb.append("<div id=\"Menu\">\n");
        sb.append("<a ID=\"Home\" href=\"Default.jsp\">首页<BR /><span class= \"Menu2\"
        >Home</span></a>");
        int i;
        Vector MenuVector=new Vector();
        MenuVector=MenuManage.getMenus(0);
        for (i=0; i<MenuVector.size(); i++) {
            MenutempMenu=(Menu) MenuVector.get(i);
            if (tempMenu.getHide()==0)
                sb.append("<a href=\"List.jsp?MenuID="
                  +tempMenu.getMenuID()+"\">"
                  +tempMenu.getMenuName()
                  +"<BR /><span class=\"Menu2\">"
                  +tempMenu.getMenuEName()+"</span></a>");
        }
        sb.append("</div>\n");
        sb.append("</div>\n");
        return sb.toString();
    } catch (SQLException e) {
        e.printStackTrace();
        return null;
    }
}
```

（4）在 Config 类中添加如程序 7-12 所示的代码。

程序 7-12 是将网页的尾部代码进行封装，使用了一个 StringBuffer 将代码封装，最后利用 StringBuffer 的 toString( )方法将代码返回，这样就对页面尾部的信息进行了处理。

【程序 7-12】 PrintFooter( )方法

Example Source Code：

```
/*****************************************************************************
 * 函数名：PrintFooter
```



```
 * 作用：封装网页尾部代码
 * 参数：无
 * 返回值：字符型。返回网页的尾部代码
 * @throws Exception
*****************************************************************************************/
public String PrintFooter() {
    try {
        StringBuffer sb=new StringBuffer();
        sb.append("<div ID=\"Footer\">\n");
        sb.append("<div ID=\"Copyright\">\n");
        sb.append("<div>\n");
        sb.append("<a href=\"About.jsp\">关于我们</a> |\n");
        sb.append("<a href=\"Readme.jsp\">安装说明</a> |\n");
        sb.append("<a href=\"Profile.jsp\">功能简介</a> |\n");
        sb.append("<a href=\"Copyright.jsp\">版权说明</a> |\n");
        sb.append("<a href=\"Admin_Main.jsp\">管理入口</a>\n");
        sb.append("</div>\n");
        sb.append("<div>Copyright &copy; 1998-2008 All Rights Reserved</div>\n");
        sb.append("<div>版权所有：@工作室</div>\n");
        sb.append("</div>\n");
        sb.append("</div>\n");
        sb.append("</body>\n");
        sb.append("</html>\n");
        return sb.toString();
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}
```

任务 5：编写表现层的 JSP 页面框架结构

“新闻类别网站”中不仅有模块化的类方法，而且表现层的 JSP 页面都采用了统一的公共框架模块结构，使上述模块如页面的头、尾及导航部分的模块化更好地体现了出来，从而实现了代码的优化和重用，在本系统中涉及表现层 JSP 页面的都将采用本框架结构，具体结构如程序 7-13。具体操作步骤如下：

（1）在 Eclipse 中新建一个 JSP 文件，作为表现层的 JSP 页面框架结构，保存到项目中，文件名为 Temp-jsp.jsp，具体步骤可参看 5.6 节的任务 2。

（2）在上步新建的 JSP 文件中添加如程序 7-13 所示的代码。

【程序 7-13】 JSP 页面的公共框架结构

Example Source Code：

```
<%@ page contentType="text/html;charset=GB2312" pageEncoding="GBK"%>
<%@ page import="com.news.*"%>
<%@ page import="com.news.menu.*"%>
<%@ page import="com.news.util.*"%>
<%@ page import="com.news.article.*"%>
<%@ page import="com.news.user.*"%>
<%@ page import="java.util.Vector"%>
```



```
<%
try{
    response.setHeader("Pragma","No-cache");
    response.setHeader("Cache-Control","no-cache");
    response.setDateHeader("Expires", 0);
    request.setCharacterEncoding("GBK");
    Config con=new Config();
    out.print(con.PrintHead("登录页面"));
    //处理代码
    //处理代码
    //处理代码
    out.print(con.PrintFooter());
}
catch(Exception e){
    e.printStackTrace();
    String errMsg=e.getMessage();
}
%>
```

7.1.3　相关知识拓展

1. 模块化设计(Block-based design)

模块化设计,简单地说就是程序的编写不是开始就逐条录入计算机语句和指令,而是首先用主程序、子程序、子过程等框架把软件的主要结构和流程描述出来,并定义和调试好各个框架之间的输入、输出接口关系。

在软件主要结构和流程的基础上,再用逐步求精的方法得到一系列以功能模块为单位的算法描述。以功能模块为单位进行程序设计,实现其求解算法的方法称为模块化。模块化的目的是为了降低程序复杂度,使程序设计、调试和维护等操作简单化。

2. 使用模块化程序设计方法

在设计程序求解问题时,首先要对问题从整体的角度进行分析,将其分解成几个有机的组成部分,如果某些部分还比较复杂可再分解,经过逐步分解和细化后,将一个大而复杂的问题从总体到局部,逐步分解为若干个小的可解的基本问题,再通过求解这些基本问题最终求解得原问题的解。

模块化程序设计方法反映了结构化程序设计的"自顶而下,逐步求精"的基本思想。

3. 提高模块的独立性

尽量使每个模块都具有独立的功能。设计程序系统时使用按功能划分模块的方法,使模块的内聚度强,耦合度弱(独立性比较高)。

4. 按层次结构组织模块

采用模块化方法得到的系统是由互相连接的模块构成的。进行模块化程序设计时,提



倡按层次结构组织各模块。

7.1.4 经验总结

作为一名网站设计者，一定要记住 package 和 import 关键字的作用就是分割单个全局命名空间，保证不会遇到名字的冲突——无论有多少人使用因特网，也无论有多少人用 Java 编写自己的类。

大家或许已注意到这样一个事实：由于一个包永远不会真的"封装"到单独一个文件里面，它可由多个.class 文件构成，所以局面可能稍微有些混乱。为避免这个问题，最合理的一种做法就是将某个特定包使用的所有.class 文件都置入单个目录里。也就是说，要利用操作系统的分级文件结构避免出现混乱局面，这正是 Java 所采取的方法。

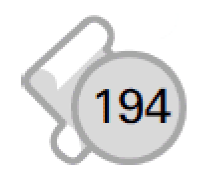
Java 同时也解决了另外两个问题：创建独一无二的包名以及找出那些可能深藏于目录结构某处的类。为达到这个目的，需要将.class 文件的位置路径编码到 package 的名字里。但根据约定，编译器强迫 package 名的第一部分是类创建者的因特网域名。由于因特网域名肯定是独一无二的(由 InterNIC 保证——注释②，它控制着域名的分配)，所以假如按这一约定行事，package 的名称就肯定不会重复，所以永远不会遇到名称冲突的问题。换句话说，除非将自己的域名转让给其他人，而且对方也按照相同的路径名编写 Java 代码，否则名字的冲突是永远不会出现的。当然，如果没有自己的域名，那么必须创造一个非常生僻的包名(例如自己的英文姓名)，以便尽最大可能创建一个独一无二的包名。如决定发行自己的 Java 代码，那么强烈推荐去申请自己的域名，它所需的费用是非常低廉的。

这部分技巧的另一方面是 Java 将 package 名解析成自己机器上的一个目录。这样一来，Java 程序运行并需要装载.class 文件的时候(这是动态进行的，在程序需要创建属于那个类的一个对象，或者首次访问那个类的一个 static 成员时)，就可以找到.class 文件驻留的那个目录。

Java 解释器的工作程序如下：首先，它找到环境变量 CLASSPATH(将 Java 或者具有 Java 解释能力的工具——如浏览器——安装到机器中时，通过操作系统进行设定)。CLASSPATH 包含了一个或多个目录，它们作为一种特殊的"根"使用，从这里展开对.class 文件的搜索。从根开始，解释器会寻找包名，并将每个点号(句点)替换成一个斜杠，从而生成从 CLASSPATH 根开始的一个路径名(所以 package foo.bar.baz 会变成 foo\bar\baz 或者 foo/bar/baz；具体是正斜杠还是反斜杠由操作系统决定)。随后将它们连接到一起，成为 CLASSPATH 内的各个条目(入口)。以后搜索.class 文件时，就可从这些地方开始查找与准备创建的类名对应的名字。此外，它也会搜索一些标准目录——这些目录与 Java 解释器驻留的地方有关。

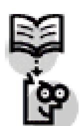
为进一步理解这个问题，下面以笔者自己的域名 bruceeckel.com 为例，将其反转过来后，com.bruceeckel 就为类创建了独一无二的全局名称(com、edu、org、net 等扩展名以前在 Java 包中都是大写的，但自 Java 1.2 以来，这种情况已发生了变化。现在整个包名都是小写的)。由于决定创建一个名为 util 的库，这里进一步分割它，最后得到的包名如下：

```
package com.bruceeckel.util;
```



现在,可将这个包名作为下述两个文件的“命名空间”使用。

7.2　实现主要功能模块



知识储备

1. static 修饰符简介

static 表示“全局”或者“静态”的意思,用来修饰成员变量和成员方法,也可以形成静态 static 代码块。

被 static 修饰的成员变量和成员方法独立于该类的任何对象。也就是说,它不依赖类特定的实例,被类的所有实例共享。只要这个类被加载,Java 虚拟机就能根据类名在运行时数据区的方法区内找到它们。因此,static 对象可以在它的任何对象创建之前访问,无需引用任何对象。

用 public 修饰的 static 成员变量和成员方法本质是全局变量和全局方法,当对它进行类的实例化时,不生成 static 变量的副本,而是类的所有实例共享同一个 static 变量。

static 变量前可以有 private 修饰,表示这个变量可以在类的静态代码块中,或者类的其他静态成员方法中使用(当然也可以在非静态成员方法中使用),但是不能在其他类中通过类名来直接引用。简单地理解,private 是访问权限限定,static 表示不要实例化就可以使用。static 前面加上其他访问权限关键字的效果也以此类推。

static 修饰的成员变量和成员方法习惯上称为静态变量和静态方法,可以直接通过类名来访问,访问语法为:

```
类名.静态方法名(参数列表……)
类名.静态变量名
```

用 static 修饰的代码块表示静态代码块,当 Java 虚拟机(JVM)加载类时,就会执行该代码块。

(1) static 变量

按照是否静态对类成员变量进行分类,可分两种:一种是被 static 修饰的变量,叫静态变量或类变量;另一种是没有被 static 修饰的变量,叫实例变量。

静态变量在内存中只有一个拷贝(节省内存),JVM 只为静态变量分配一次内存,在加载类的过程中完成静态变量的内存分配,可用类名直接访问,也可以通过对象来访问。

实例变量每创建一个实例,就会为实例变量分配一次内存,实例变量可以在内存中有多个副本,互不影响。

(2) 静态方法

静态方法可以直接通过类名调用,任何实例都可以调用,因此静态方法中不能用 this 和 super 关键字,不能直接访问所属类的实例变量和实例方法(即不带 static 的成员变量和成员方法),只能访问所属类的静态成员变量和成员方法。

因为 static 方法独立于任何实例,因此 static 方法必须被实现,而不能是抽象的(abstract)。



(3) static 代码块

static 代码块也叫静态代码块,是在类中独立于类成员的 static 语句块,可以有多个,位置可以随便放,它不在任何方法体内,JVM 加载类时会执行这些静态的代码块。如果 static 代码块有多个,JVM 将按照它们在类中出现的先后顺序依次执行,每个代码块只会被执行一次。

利用静态代码块可以对一些 static 变量进行赋值,对于 static 类型的 main 方法,JVM 在运行 main 方法的时候可以直接调用而不用创建实例。

(4) static 和 final

static final 用来修饰成员变量和成员方法,可简单理解为"全局常量"。

static final 用于变量,表示一旦赋值就不可修改,并且通过类名可以访问。

static final 用于方法,表示此方法不可覆盖,并且可以通过类名直接访问。

2. Vector 类

Vector 类被称为向量,用来创建可包含多个对象的动态数组,它提供加入、删除及插入元素的方法。创建一个 vector 类时,必须指出它的初始大小及增量。

(1) 构造 Vector 类的三种方式

public Vector( ):创建空的向量。

public Vector(int initCap):创建由 initCap 指出初始容量的向量。

public Vector(int initCap, int increment):创建由 initCap 指出的元素空间的向量并通过增量元素增加容量。

(2) 在 Vector 中加入对象

addElement(Object n):新对象 n 作为最后一个元素加入 vector 中。

insertElementAt(Object n, int index):在下标所指的位置插入一个新的对象 n。

setElementAt(Object n, int index):用对象 n 在下标处替换此元素。

(3) 在 Vectors 中删除对象

可用 void removeElementAt(int index)在 vector 中删除对象。

(4) 访问 Vector 中元素

elementAt( )方法用于访问向量中的元素;size( )方法返回 vector 中对象的个数。

注意

Vector 类提供了可以动态增加的存储空间。它和数组的一个明显区别是:数组中的元素只能是同一种类型,Vector 中可以存入任意类型。

7.2.1 任务描述

利用 Eclipse 实现"新闻类别网站"的主要功能模块。

7.2.2 任务实现

根据第 2 章"新闻类别网站"的功能设计要求,在此分三个任务实现该网站的主要功能。

任务 1 栏目管理功能模块的实现



任务 2　新闻管理功能模块的实现

任务 3　管理员功能模块的实现

任务 1：实现栏目管理功能模块

栏目管理功能模块是整个系统的结构层，处于整个系统的中层，主要实现的是对系统中栏目的动态管理，即在整个系统的设计中，它是可以随时变化的。为方便实现，根据功能模块实现的先后顺序将整个栏目管理功能模块的实现划分为如下三个更具体的子任务来完成，实际上所有功能模块的实现都可按照这样的划分进行。

子任务 1.1　构建包结构

子任务 1.2　建立所需类的基本结构

子任务 1.3　表现层的实现

子任务 1.1：构建包结构

在 Eclipse 中新建“com. news. menu”包结构，详细操作步骤参见 7.1 中“任务实现”部分的任务 1。

子任务 1.2：建立所需类的基本结构

具体操作步骤如下：

（1）在本节子任务 1.1 中建立的“com. news. menu”包中新建基本类 Menu，保存为 Menu. java。其中的字段与数据库中对应的实体数据表中的字段名一致，每个字段都要编写 getXxx 和 setXxx 方法，大致写法如程序 7-14 所示。

【程序 7-14】　基本类 Menu

Example Source Code：

```
package com.news.menu;

/* *
 * Title: @工作室
 * Description: Copyright:
 * Copyright (c) 2002 Company:
 * www.uc2008.net
 *
 * @author: Skimob
 * @version 1.0
 * /
import java.sql.ResultSet;
import com.news.database.DBConnect;
import com.news.util.* ;
public class Menu {
    …
    int MenuID;
    …
    public Menu( ) {
    }
    …
    public void setMenuID(int MenuID) {
        this.MenuID=MenuID;
    }
```



```
        public int getMenuID( ) {
            return this.MenuID;
        }
        …
    }
```

（2）在本节子任务 1.1 中建立的“com. news. menu”包中新建类 MenuManage，保存为 MenuManage. java。完善所需要的方法，即：新闻的列表显示、增加、修改、删除的操作，大致写法如程序 7-15 所示。

该类的功能用于管理员对新闻系统版面的维护。包括了对新闻系统版面的增删改功能，添加一个新版面是调用函数 saveMenu，修改新闻时调用函数 saveEdit，删除一个新闻时调用函数 delMenu。

【程序 7-15】 MenuManage 类

Example Source Code：

```
package com.news.menu;

/* *
 * Title:          @工作室
 * Description:
 * Copyright:      Copyright (c) 2002
 * Company:        www.uc2008.net
 * @author:        Skimob
 * @version 1.0
 * /
import java.sql.ResultSet;
import java.util.Vector;
import javax.servlet.http.HttpServletRequest;
import com.news.database.DBConnect;
import com.news.article.* ;
import com.news.util.* ;
public class MenuManage {
    public static Vector getMenu( ) throws MenuNotFoundException {}
    public static Vector getMenus(int MenuID) throws Exception {}
    public static Menu getMenu(int MenuID) throws Exception {}
    public static void OrdersMenu(HttpServletRequest request) throws Exception {}
    public static void saveMenu(HttpServletRequest request) throws Exception {}
    public static void saveEdit(HttpServletRequest request) throws Exception {}
    public static void delMenu(int MenuID) {}
    public static boolean hideMenu(HttpServletRequest request){}
    public static void updateOrders(HttpServletRequest request)throws Exception {}
}
```

子任务 1.3：表现层的实现

栏目管理表现层的 JSP 页面将采用 7.1 节任务 5 中完成的表现层 JSP 框架结构为基础，将所有对系统栏目的管理操作集中在这个页面。如：系统栏目的增、删、改、查等操作，具体结构如程序 7-16 所示。具体操作步骤如下：

（1）在 Eclipse 中新建一个 JSP 文件，作为系统栏目管理的 JSP 页面，保存到项目中，文



件名为 Admin_Menu.jsp，具体步骤可参看 5.6 节的任务 2。

（2）在新建的 JSP 文件中添加如程序 7-16 所示的代码。

【程序 7-16】　系统栏目管理模块表现层的实现

Example Source Code：

```
<%@page contentType="text/html;charset=GB2312" pageEncoding="GBK"%>
<%@page import="com.news.*"%>
<%@page import="com.news.menu.*"%>
<%@page import="com.news.util.*"%>
<%@page import="com.news.article.*"%>
<%@page import="com.news.*,com.news.menu.*,com.news.util.*,com.news.article.*"%>
<%@page import="java.util.Vector"%>
<%@include file="Session.jsp"%>
<%
try{
    Function Fun=new Function();
    Config con=new Config();
    out.print(con.PrintHead(""));
    String IP=request.getRemoteAddr();                //得到客户端 IP 地址
    //String AdminName=(String)session.getAttribute("AdminName");
    String AdminName="admin";
    String Action=ParamUtil.getString(request,"Action","");
    int MenuID=ParamUtil.getInt(request,"MenuID",1);
%>
<div id="Main">
<div id="Main_left" class="list_list">
<div class="List">
<div class="title">
    <div class="title_left">导航菜单</div>
    <div class="title_right">@工作室</div>
    <div class="clear"></div>
</div>
<div class="connect">
<%
    out.print(con.Admin_Menu(request));
    %>
</div>
</div>
</div>
<div id='Main_right' class="list_main">
<div class="List">
<div class="title">
<div class="title_left">热点关注</div>
<div class="title_right"><a href="">@工作室</a></div>
<div class="clear"></div>
</div>
<div class="connect">
<%
        //保存添加的分类
        if (Action.equals("save"))
```



```
    {
        MenuManage.saveMenu(request);
        out.print("<script>alert('添加分类操作成功!');location.href='Admin_Class.
        jsp';</script>");
    }
    //保存修改的分类
    else if (Action.equals("saveedit"))
    {
        MenuManage.saveEdit(request);
        out.print("<script>alert('修改分类操作成功!');location.href='Admin_Class.
        jsp';</script>");
    }
    //删除分类
    else if (Action.equals("del"))
    {
        MenuManage.delMenu(MenuID);
        out.print("<script>alert('删除分类操作成功!');location.href='Admin_Class.
        jsp';</script>");
    }
    //分类隐藏
    else if (Action.equals("hide"))
    {
        if (MenuManage.hideMenu(request)){
    out.print("<script>location.href='Admin_Class.jsp';</script>");
        }
        else
    out.print("<script>alert('隐藏操作失败!');location.href='Admin_Class.jsp';</
    script>");
    }
    //分类排序
    else if (Action.equals("orders"))
    {
        MenuManage.OrdersMenu(request);
        out.print("<script>alert('排序操作成功!');location.href='Admin_Class.jsp';
        </script>");
    }
    //添加分类
    else if (Action.equals("add"))
    {
⋮
}
//编辑分类
else if (Action.equals("edit"))
{
      Menu sa=(Menu)MenuManage.getMenu(MenuID);
    ⋮
}
//读取类别列表
else
{
⋮
```






```
        }
        %>
    </div>
    </div>
    </div>
    <div class="clear"></div>
    </div>
    <%
    out.print(con.PrintFooter());
    }
    catch(Exception e){
        e.printStackTrace();
        String errMsg=e.getMessage();
    }
    %>
```

任务 2：新闻管理功能模块的实现

一般的新闻组件维护需要的操作有增加新闻、修改新闻、删除新闻、查询所有的新闻以及查询某一个新闻的属性数据。“新闻类别网站”这些功能的实现具体分三个子任务来实现。

子任务 2.1　构建包结构

子任务 2.2　建立所需类的基本结构

子任务 2.3　表现层的实现

子任务 2.1：构建包结构

在 Eclipse 中新建“com. news. article”包结构，详细操作步骤参见 7. 1 节中“任务实现”部分的任务 1。

子任务 2.2：建立所需类的基本结构

具体操作步骤如下：

(1) 在本节任务 2. 1 中建立的“com. news. article”包中新建基本类 Article，保存为 Article. java。其中的字段与数据库中对应的实体数据表中的字段名一致，每个字段都要编写 getXxx 和 setXxx 方法，大致写法如程序 7-17 所示。

【程序 7-17】　基本类 Article

Example Source Code：

```
package com.news.article;
import com.news.util.*;
public class Article
{
    ⋮
    String Title;
    ⋮
    public Article(){}
    ⋮
    public String getTitle(){
        return this.Title;
    }
```



```
    public void setTitle(String title){
        this.Title=title;
    }
    ⋮
}
```

（2）在本节任务 2.1 中建立的“com. news. article”包中新建类 ArticleManage，保存为 ArticleManage. java，并完善所需要的方法，即：对新闻进行列表显示、增加新闻、修改新闻、删除新闻，大致写法如程序 7-18 所示。

【程序 7-18】 ArticleManage 类

Example Source Code：

```
package com.news.article;

/* *
 * Title:          @工作室
 * Description:
 * Copyright:      Copyright (c) 2002
 * Company:        www.uc2008.net
 * @author:        Skimob
 * @version 1.0
 * /

import java.sql.ResultSet;
import java.util.Vector;
import javax.servlet.http.HttpServletRequest;

import com.news.database.DBConnect;
import com.news.menu.Menu;
import com.news.menu.MenuManage;
import com.news.util.Format;
import com.news.util.ParamUtil;

public class ArticleManager {

    public static Article getArticle(int ArticleID) throws ArticleNotFoundException {}
     public static Vector getArticles (int MenuID, String type) throws ArticleNotFound
     Exception {}
    public static Vector getArticles( ) throws ArticleNotFoundException {}
    public static Vector getArticles(int MenuID) throws Exception {}
    public static void saveArticle(HttpServletRequest request) throws Exception {}
    public static void saveEdit(HttpServletRequest request) throws Exception {}
    public static void delArticles(HttpServletRequest request) throws Exception {}
    public static void updateHit(HttpServletRequest request)throws Exception {}
}
```

子任务 2.3：表现层的实现

新闻管理表现层的 JSP 页面以 7.1 节的任务 5 中完成的表现层 JSP 框架结构为基础，将系统中所有栏目下的新闻管理操作（新闻的增、删、改、查等）集中在这个页面。具体结构



如程序 7-19 所示。具体操作步骤如下：

（1）在 Eclipse 中新建一个 JSP 文件，作为系统栏目管理的 JSP 页面，保存到项目中，文件名为 Admin_Articles.jsp，具体步骤可参看 5.6 节的"任务 2"。

（2）在新建的 JSP 文件中添加如程序 7-19 所示的代码。

【程序 7-19】 新闻管理模块表现层的实现

Example Source Code：

```
<%@page contentType="text/html;charset=GB2312" pageEncoding="GBK"%>
<%@page import="com.news.*"%>
<%@page import="com.news.menu.*"%>
<%@page import="com.news.util.*"%>
<%@page import="com.news.article.*"%>
<%@page import="com.news.user.*"%>
<%@page import="java.util.Vector"%>
<%@taglib uri="WEB-INF/FCKeditor.tld" prefix="FCK"%>
<%@include file="Session.jsp"%>
<%
    try {
        Config con=new Config();
        out.print(con.PrintHead(""));

        String Action=ParamUtil.getString(request, "Action");
        int ID=ParamUtil.getInt(request, "ID", 0);
        int ClassID=ParamUtil.getInt(request, "ClassID", 0);
%>
<script type="text/javascript" src="/FCKeditor/fckeditor.js"></script>
<div id="Main">
<div id="Main_left" class="list_list">
<div class="List">
<div class="title">
<div class="title_left">导航菜单</div>
<div class="title_right">@工作室</div>
<div class="clear"></div>
</div>
<div class="connect">
<%
    out.print(con.Admin_Menu(request));
%>
</div>
</div>
</div>
<div id='Main_right' class="list_main">
<div class="List">
<div class="title">
<div class="title_left">热点关注</div>
<div class="title_right"><a href="">@工作室</a></div>
<div class="clear"></div>
</div>
<div class="connect">
```



```
<%
    if (Action==null || Action.equals(""))
            Action="list";

    if (Action.equals("list")) {
        Vector news;
        if (ClassID==0)
            news=ArticleManager.getArticles();
        else
            news=ArticleManager.getArticles(ClassID);
        //
    } else if (Action.equals("edit")) {
        Article theNew= (Article) ArticleManager.getArticle(ID);
        //
    } else if (Action.equals("add")) {
        //
    } else if (Action.equals("del")) {
        ArticleManager.delArticles(request);
        out.println("<SCRIPT LANGUAGE='JavaScript'>alert('del 操作成功!');location.href='
        Admin_News.jsp';</SCRIPT>");
    } else if (Action.equals("saveedit")) {
        ArticleManager.saveEdit(request);
        out.println("saveedit 操作成功!");
        out.println("<SCRIPT LANGUAGE='JavaScript'>alert('del 操作成功!');location.href='
        Admin_News.jsp';</SCRIPT>");
    } else if (Action.equals("save")) {
        ArticleManager.saveArticle(request);
        out.println("save 操作成功!");
        out.println("<SCRIPT LANGUAGE='JavaScript'>alert('del 操作成功!');location.href='
        Admin_News.jsp';</SCRIPT>");
    } else {
        out.println("其他操作成功!");
    }
%>
</div>
</div>
</div>
<div class="clear"></div>
</div>
<%
    out.print(con.PrintFooter());
    } catch (Exception e) {
        e.printStackTrace();
        String errMsg=e.getMessage();
    }
%>
```

注意：此处并没有将所有的代码给出，只是给出了该页面的基本框架结构，具体的代码可参考“新闻类别网站”源文件。





任务 3：用户管理功能模块的实现

用户管理功能模块维护需要的操作有增加用户、修改用户、删除用户、查询所有的用户、查询某一个用户的属性数据。具体分三个子任务来实现。

子任务 3.1　构建包结构
子任务 3.2　建立所需类的基本结构
子任务 3.3　表现层的实现

子任务 3.1：构建包结构

在 Eclipse 中新建“com. news. user”包结构，详细操作步骤参见 7.1 节中“任务实现”部分的任务 1。

子任务 3.2：建立所需类的基本结构

具体操作步骤如下：

(1) 在本节任务 3.1 中建立的“com. news. user”包中新建基本类 User，保存为 User. java。其中的字段与数据库中对应的实体数据表中的字段名一致，每个字段都要编写 getXxx 和 setXxx 方法，大致写法如程序 7-20 所示。

【程序 7-20】 基本类 User

Example Source Code：

```
package com.news.user;

/* *
 * Title:         @工作室
 * Description:
 * Copyright:     Copyright (c) 2002
 * Company:       www.uc2008.net
 * @author:       Skimob
 * @version 1.0
 * /

import java.sql.ResultSet;

import com.news.database.DBConnect;
import com.news.menu.Menu;
import com.news.util.Format;

public class User{
    …
    int ID;
    String Name;
    String Pwd;
    int Type;
    …
    public User( ){
    }
    public User(int ID){}
    …
    public void setID(int ID){
```



```
        this.ID= ID;
    }
    public int getID( ){
        return this.ID;
    }
    …
}
```

（2）在本节任务 3.1 中建立的“com.news.user”包中新建类 UserManage，保存为 UserManage.java，并完善所需要的方法，即：对系统用户进行列表显示、增加新用户、修改用户、删除用户的操作，大致写法如程序 7-21 所示。

【程序 7-21】 UserManage 类

Example Source Code：

```
package com.news.user;

/* *
 * Title:          @工作室
 * Description:
 * Copyright:      Copyright (c) 2002
 * Company:        www.uc2008.net
 * @author:        Skimob
 * @version 1.0
 * /

import java.sql.ResultSet;
import java.util.Vector;
import javax.servlet.http.HttpServletRequest;

import com.news.database.DBConnect;
import com.news.util.* ;

public class UserManager {

    public static Vector getUsers (HttpServletRequest request) throws UserNotFound Exception
    {}
    public static User getUser (HttpServletRequest request) throws UserNotFound Exception {}
    public static void saveNew (HttpServletRequest request) throws Exception {}
    public static void saveEdit (HttpServletRequest request) throws Exception {}
    public static void delUser (HttpServletRequest request) throws Exception {}
}
```

子任务 3.3：表现层的实现

用户管理表现层的 JSP 页面以 7.1 节任务 5 中完成的表现层 JSP 框架结构为基础，将系统中所有用户的管理操作（用户的增、删、改、查）集中在这个页面。具体操作步骤如下：

（1）在 Eclipse 中新建一个 JSP 文件，作为系统用户管理的 JSP 页面，保存到项目中，文件名为 Admin_User.jsp，具体步骤可参看 5.6 节的任务 2。

（2）在新建的 JSP 文件中添加如程序 7-22 所示的代码。



【程序 7-22】 列表程序

```
<%@page contentType="text/html;charset=GB2312" pageEncoding="GBK"%>
<%@page import="com.news.*"%>
<%@page import="com.news.menu.*"%>
<%@page import="com.news.util.*"%>
<%@page import="com.news.article.*"%>
<%@page import="com.news.user.*"%>
<%@page import="java.util.Vector"%>

<%@include file="Session.jsp"%>
<%
    try{
    Config con=new Config();
    out.print(con.PrintHead(""));
%>
<div id="Main">
<div id="Main_left" class="list_list">
<div class="List">
<div class="title">
<div class="title_left">导航菜单</div>
<div class="title_right">@工作室</div>
<div class="clear"></div>
</div>
<div class="connect">
<%
    out.print(con.Admin_Menu(request));
%>
</div>
</div>
</div>
<div id='Main_right' class="list_main">
<div class="List">
<div class="title">
<div class="title_left">热点关注</div>
<div class="title_right"><a href="">@工作室</a></div>
<div class="clear"></div>
</div>
<div class="connect">
<%
    String Action=ParamUtil.getString(request,"Action","");
    int UserID=ParamUtil.getInt(request,"UserID",0);
    String UserName=ParamUtil.getString(request,"UserName","");

    if (Action==null || Action.equals("")) Action="list";
    if (Action.equals("list"))
    {
        Vector users;
        users=UserManager.getUsers(request);
        int orders=ParamUtil.getInt(request,"orders",7);
        int usersNum=users.size();
```



```
            int Page=ParamUtil.getInt(request,"Page",1);

        }
        else if (Action.equals("edit"))
        {
            User theUser= (User)UserManager.getUser(request);

        }
        else if (Action.equals("add"))
        {

        }
        else if(Action.equals("del"))
        {
            UserManager.delUser(request);
            out.print("<script>alert('del 操作成功！');location.href='Admin_User.jsp';</
            script>");
        }
        else if(Action.equals("save"))
        {
            UserManager.saveNew(request);
            out.println("save 操作成功！");
        }
        else if(Action.equals("saveedit"))
        {
            UserManager.saveEdit(request);
            out.println("saveEdit 操作成功！");
        }
        else
        {
            //out.println(UserID);

            //UserManager.delUser(request);
            out.println("其他操作成功！");
        }
%>
</div>
</div>
</div>
<div class="clear"></div>
</div>
<%
    out.print(con.PrintFooter());
}
catch(Exception e){
    e.printStackTrace();
    String errMsg=e.getMessage();
}
%>
```





7.2.3　相关知识拓展

1. Java Servlet 与 JSP 的比较

Java Server Pages(JSP)是一种实现普通静态 HTML 和动态 HTML 混合编码的技术，JSP 并没有增加任何本质上不能用 Servlet 实现的功能。但是，在 JSP 中编写静态 HTML 更加方便，不必再用 println 语句来输出每一行 HTML 代码。更重要的是，借助内容和外观的分离，页面制作中不同性质的任务可以方便地分开，比如，由页面设计者进行 HTML 设计，同时留出供 Servlet 程序员插入动态内容的空间。

2. Servlet 的一些特点

(1) 高效

在服务器上仅有一个 Java 虚拟机在运行，它的优势在于当 Servlet 被客户端发送的第一个请示激活以后，将继续运行到后台，等待以后的请求，每个请求将生成一个线程而不是进程。

(2) 方便

Servlet 提供了大量的实用工具例程，例如处理很难完成的 HTML 表单数据读取、设置 HTTP 头处理、Cookie 跟踪会话。

(3) 可移植性好

Servlet 用 Java 写 Servlet API 具有完善的标准。现在有很多企业编写 Servlet，使得它无需任何实质上的改动即可移植到 Apache 和 Micorosoft IIS 上。

(4) 节省投资

Servlet 建站很简单，用钱不多就可实现。

3. JSP 的开发模式

JSP 自产生到现在，应用越来越广泛，其相关技术也越来越多，如 JavaBean、EJB 等。相关技术的产生，使 JSP 技术更容易实现 Web 网站的开发和控制。JSP 网站开发经常使用下面几种组合方式来进行，包括纯粹 JSP 技术实现、JSP＋JavaBean 实现、JSP＋JavaBean＋Servlet 实现、J2EE 实现等。不同的开发组合，可以称为不同的设计模式，最常用的技术是 JSP＋JavaBean＋Servlet。

(1) 纯粹 JSP 实现

使用纯粹 JSP 技术实现动态网站开发，是 JSP 初学者经常使用的技术。JSP 页面中所有的代码都是在同一个页面上，如＜html＞标记、＜css＞标记、＜javascript＞标记、逻辑处理、数据库处理代码等。这么多代码混合在一个页面中，容易出现错误，出现错误后，不容易查找和调试。这样设计出的网站，采用 JSP 技术和采用 ASP 技术没有多大的差别。

(2) JSP＋JavaBean 实现

JSP＋JavaBean 技术的使用，很好地实现了网页静态部分和动态部分相互分离。在这种技术中，使用 JSP 技术中的 HTML、CSS 等可以非常容易地构建数据显示页面，而对于数据处理，可以交给 JavaBean 技术处理，如连接数据库代码、显示数据库代码。当执行功能代



码封装到 JavaBean 中时，同时也达到了代码重用的目的。如显示当前时间的 JavaBean，不仅可以用在当前页面，还可以用在其他页面。这种技术的使用，已经显示出 JSP 技术的优势，但并不能充分展现 JSP 的优势，JSP＋JavaBean＋Servlet 技术的组合更加充分地显示了 JSP 的优势。

(3) JSP＋JavaBean＋Servlet 实现

JSP＋JavaBean＋Servlet 技术的组合，很好地实现了 MVC 模式。MVC 模式是 Model-View-Controller 的缩写，中文翻译为“模式-视图-控制器”，MVC 应用程序由这三个部分组成。Event（事件）导致 Controller 改变了 Model 或 View，或者同时改变两者。只要 Controller 改变了 Models 的数据或者属性，所有依赖的 View 都会自动更新。类似的，只要 Controller 改变了 View，View 会从潜在的 Model 中获取数据来刷新自己。MVC 模式最早是 Smalltalk 语言研究团提出的，应用于用户交互应用程序。Smalltalk 语言和 Java 语言有很多相似性，都是面向对象语言。

MVC 模式是一个复杂的架构模式，其实现也显得非常复杂。但是，人们已经总结出了很多可靠的设计模式，多种设计模式结合在一起，使 MVC 模式的实现变得相对简单易行。Views 可以看做一棵树，显然可以用 Composite Pattern 来实现。Views 和 Models 之间的关系可以用 Observer Pattern 来实现。Controller 控制 Views 的显示，可以用 Strategy Pattern 实现。Model 通常是一个调停者，可采用 Mediator Pattern 来实现。

现在来了解一下 MVC 三个部分在架构中处于什么位置，这样有助于理解 MVC 模式的实现。MVC 与架构的对应关系是：View 处于 Web Tier 或者是 Client Tier，通常是 JSP/Servlet，即页面的显示部分。Controller 也处于 Web Tier，通常用 Servlet 来实现，即页面显示的逻辑部分的实现。Model 处于 Middle Tier，通常用服务器端的 JavaBean 或者 EJB 实现，即业务逻辑部分的实现。其形式如图 7-3 所示。

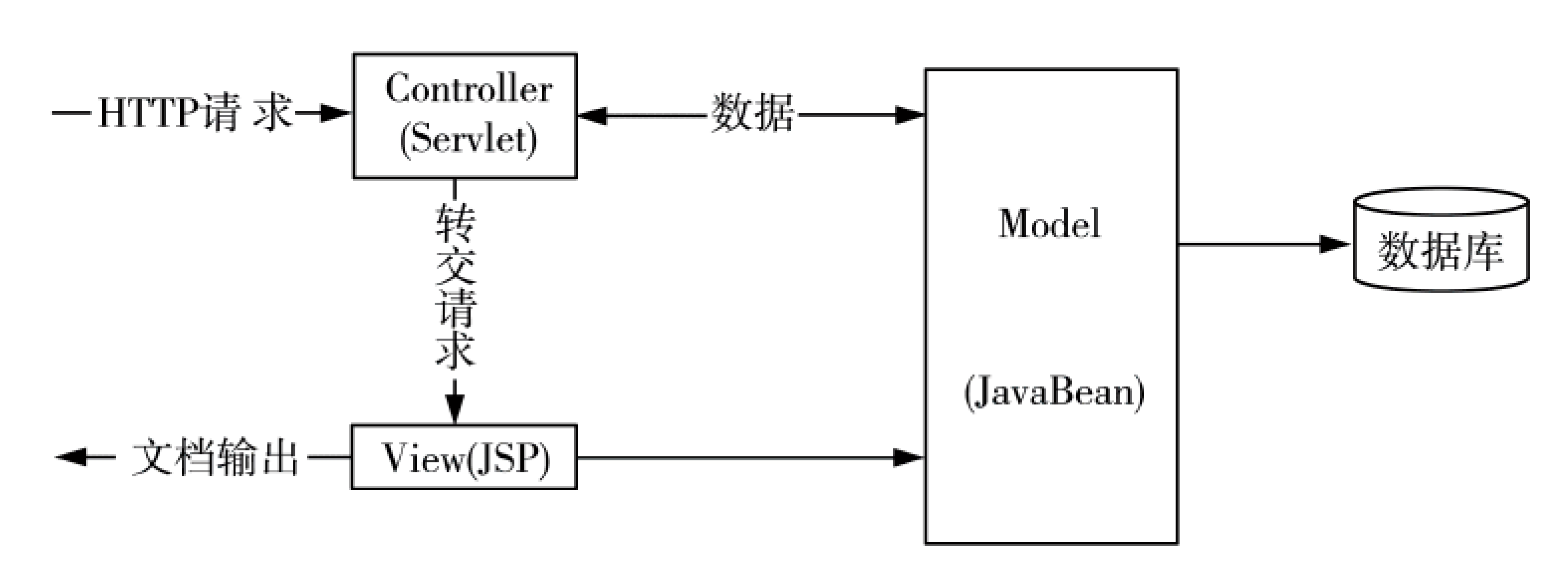


图 7-3　MVC 三个部分的形式

视图（View）代表用户交互界面，对于 Web 应用来说，可以概括为 HTML 界面，但有可能为 XHTML、XML 和 Applet。随着应用的复杂性和规模性，界面的处理也变得具有挑战性。一个应用可能有很多不同的视图，MVC 设计模式对于视图的处理仅限于视图上数据的采集和处理，以及用户的请求，而不包括在视图上的业务流程的处理。业务流程的处理交给模型（Model）来处理。比如一个订单的视图只接受来自模型的数据，并显示给浏览者，以及将浏览者界面的输入数据和请求传递给控制和模型。

模型（Model）是业务流程/状态的处理，以及业务规则的制定。业务流程的处理过程对其他层来说是暗箱操作，模型接受视图请求的数据，并返回最终的处理结果。业务模型的设计可以说是 MVC 最主要的核心。通过 MVC 设计模式可知，把应用的模型按一定的规则

抽取出来，抽取的层次很重要。抽象与具体不能隔得太远，也不能太近。

控制器(Controller)可以理解为接受浏览者的请求，将模型与视图匹配在一起，共同完成浏览者的请求。控制层的作用就像一个分发器，选择什么样的模型，选择什么样的视图，可以完成什么样的浏览者请求，控制层并不做任何的数据处理。例如，浏览者点击一个链接，控制层接受请求后并不处理业务信息，它只把浏览者的信息传递给模型，告诉模型做什么，选择符合要求的视图返回给浏览者。因此，一个模型可能对应多个视图，一个视图可能对应多个模型。

模型、视图与控制器的分离，使得一个模型可以具有多个显示视图。如果通过某个视图的控制器改变了模型的数据，所有其他依赖于这些数据的视图都应反映到这些变化。因此，无论何时发生了何种数据变化，控制器都会将变化通知所有的视图，导致显示的更新。这实际上是一种模型的变化—传播机制。

(4) J2EE 模式实现

J2EE 是纯粹基于 Java 的解决方案。1998 年，Sun 发布了 EJB 1.0 标准。EJB 为企业级应用中必不可少的数据封装、事务处理、交易控制等功能提供了良好的技术基础。至此，J2EE 平台的三大核心技术 Servlet、JSP 和 EJB 都已先后问世。1999 年，Sun 正式发布了 J2EE 的第一个版本。到 2003 年时，Sun 的 J2EE 版本已经升级到了 J2EE 1.4 版，其中三个关键组件的版本也演进到了 Servlet 2.4、JSP 2.0 和 EJB 2.1。至此，J2EE 体系及相关的软件产品已经成为了 Web 服务器端开发的一个强有力的支撑环境。在这种模式里，EJB 替代了前面提到的 JavaBean 技术。

在 Web 服务器端，2000 年以后出现了几种主要的技术融合方式。首先，越来越多的 Web 开发环境开始支持 MVC(Model-View-Contorller)的设计模型，为开发者提供了全套的开发框架。实际上，J2EE 和.NET 平台本身就是这种开发框架的典型代表。开源项目在 Web 开发框架和应用模型方面表现得非常积极，Struts、Jetspeed、JPortlet、Cocoon、Lenya、XOOPS 等都是开源世界里与 MVC 开发框架、门户服务和 Web 内容管理相关的优秀解决方案。

J2EE 设计模式由于框架大、不容易编写、不容易调试，因此比较难以掌握。目前只是应用在一些大型的网站上。

对于本节介绍的 4 种 JSP 开发模式，常用的是第二种和第三种。只有在充分理解这两种开发模式的基础上，才能掌握 J2EE 的设计模式。

7.2.4　经验总结

1. 向量与数组的区别

(1) Java 的数组可存储任何类型的数组元素，包括数值类型和所有类类型。

(2) Java 向量只能存储对象类的实例。

向量能重定位和调整自己的容量。默认时，向量重定位一次，其容量就扩大一倍，呈指数增长，所以应指定容量增量，作为向量构造函数的第二个参数。例如：

```
Vector itemsOrdered=new Vector(3,10)
```

此例中向量每重定位一次，其容量都增加 10 个元素。



如果向量已达到其永久大小，则可调用 tirmToSize 方法，将内存块大小调整到刚好能容纳当前个数的向量元素。

向量不能像数组一样使用[]来访问其中的元素，而必须用 elementAt 和 setSlementAt 方法来访问或修改元素。

2. 使用 Debug 调试程序

Eclipse 中设有专门的调试模式，常用的操作有：在 Java 文件中设置断点、单步调试、跳出函数、执行到最后等。

7.3 实现其他功能模块

7.3.1 任务描述

实现“新闻类别网站”的其他模块。包括：友情链接、系统日志等模块。

7.3.2 任务实现

本节的任务实现具体包括以下两个任务。

任务 1　实现友情链接模块

任务 2　实现日志设置模块

任务 1：实现友情链接模块

友情链接是一个站点常设的栏目，“新闻类别网站”中也设置了友情链接模块，该模块的具体实现分三个子任务。

子任务 1.1　构建包结构

子任务 1.2　建立所需类的基本结构

子任务 1.3　表现层的实现

子任务 1.1：构建包结构

在 Eclipse 中新建“com. news. link”包结构，详细操作步骤参见 7. 1 中“任务实现”部分的任务 1。

子任务 1.2：建立所需类的基本结构

具体操作步骤如下：

(1) 在本节任务 1.1 中建立的“com. news. link”包中新建基本类 Link，保存为 Link. java。其中的字段与数据库中对应的实体数据表中的字段名一致，每个字段都要编写 getXxx 和 setXxx 方法，大致写法如程序 7-23 所示。

【程序 7-23】 基本类 link 类

Example Source Code：

```
package com.news.link;

/* *
 * Title:         @工作室
```






```
 * Description:
 * Copyright:      Copyright (c) 2002
 * Company:        www.uc2008.net
 * @author:        Skimob
 * @version 1.0
 * /

public class Link{
      …
    int linkID;
      …
    public Link( ){
    }
      …
    public void setLinkID(int linkID){
        this.linkID=linkID;
    }
    public int getLinkID( ){
        return this.linkID;
    }
    …
}
```

（2）在本节任务 1.1 中建立的“com.news.link”包中新建类 LinkManager，保存为 LinkManager.java。完善所需要的方法，即实现友情链接的列表显示、增加、修改、删除的操作，大致写法如程序 7-24 所示。

【程序 7-24】 LinkManager 类

Example Source Code：

```
package com.news.link;

/* *
 * Title:          @工作室
 * Description:
 * Copyright:      Copyright (c) 2002
 * Company:        www.uc2008.net
 * @author:        Skimob
 * @version 1.0
 * /
import java.sql.ResultSet;
import java.util.Vector;

import javax.servlet.http.HttpServletRequest;

import com.news.database.DBConnect;
import com.news.util.ParamUtil;

public class LinkManager{

    public static Vector getLinks( ) throws LinkNotFoundException{}
```



```
    public static Link getLink(int linkID) throws Exception{}
    public static void saveNew(HttpServletRequest request) throws Exception{}
    public static void saveEdit(HttpServletRequest request) throws Exception{}
    public static void delLink (HttpServletRequest request) throws Exception{}
    public static void updateOrders(HttpServletRequest request) throws Exception{}
}
```

子任务 1.3：表现层的实现

友情链接管理表现层的 JSP 页面以本章 7.1 节任务 5 中完成的表现层 JSP 框架结构为基础，将系统中友情链接的管理操作(友情链接的增、删、改、查)集中在这个页面。

在 Eclipse 中新建一个 JSP 文件，作为友情链接管理的 JSP 页面，保存到项目中，文件名为 Admin_Link.jsp，具体步骤可参看 5.6 节的任务 2，其他的具体实现参照其他页面程序的实现，由读者自己完成。

任务 2：实现日志设置模块

日志是存储软件程序、服务或操作系统产生的消息记录的文件。在“新闻类别网站”中，也设计了系统日志模块，用来记录系统发生的事件，为系统的运行监控做好日常的记录，具体实现分三个子任务。

子任务 2.1　构建包结构
子任务 2.2　建立所需类的基本结构
子任务 2.3　表现层的实现

子任务 2.1：构建包结构

在 Eclipse 中新建“com.news. log”包结构，详细操作步骤参见 7.1 中“任务实现”部分的任务 1。

子任务 2.2：建立所需类的基本结构

具体操作步骤如下：

(1) 在本节任务 2.1 中建立的“com.news. log”包中新建基本类 Log，保存为 Log.java。其中的字段与数据库中对应的实体数据表中的字段名一致，每个字段都要编写 getXxx 和 setXxx 方法，大致写法如程序 7-25 所示。

【程序 7-25】 基本类 log 类

Example Source Code：

```
package com.news. log;

/* *
 * Title:          @工作室
 * Description:
 * Copyright:      Copyright (c) 2002
 * Company:        www.uc2008.net
 * @author:        Skimob
 * @version 1.0
 * /

public class Log{
    …
```

```
    int logID;
    …
    public Log ( ){
    }
    …
    public void setLogID(int logID){
        this. logID=logID;
    }
    public int getLogID( ){
        return this.logID;
    }
    …
}
```

（2）在本节任务 2. 1 中建立的“com. news. log”包中新建类 LogManager，保存为 LogManager. java。完善所需要的方法，即实现系统日志的列表显示、增加、修改、删除的操作，大致写法如程序 7-26 所示的代码。

【程序 7-26】 LogManager 类

Example Source Code：

```
package com.news.log;

/* *
 * Title:          @工作室
 * Description:
 * Copyright:      Copyright (c) 2002
 * Company:        www.uc2008.net
 * @author:        Skimob
 * @version 1.0
 * /
import java.sql.ResultSet;
import java.util.Vector;

import javax.servlet.http.HttpServletRequest;

import com.news.database.DBConnect;
import com.news.util.ParamUtil;

public class LogManager{
    public static Vector getLogs( ) throws LogNotFoundException{}
    public static Log getLog (int logID) throws Exception{}
    public static void saveNew(HttpServletRequest request) throws Exception{}
    public static void saveEdit(HttpServletRequest request) throws Exception{}
    public static void delLog (HttpServletRequest request) throws Exception{}
}
```

子任务 2.3：表现层的实现

参见 7.3 节子任务 1.3 的描述。



7.3.3 相关知识拓展

1. JSP 概述

JSP 是由 Sun 公司为创建动态 Web 页面而定义的一种技术。它们是与 Java 相关的一种 HTML 文档,Java 提供动态内容。JSP 是在服务器端应用的一种脚本,它接受请求并生成响应。此请求通常由一个 Web 客户端发送,而响应则是根据请求生成的一种 HTML 文档,该文档需要返回给 Web 客户端。由于 JSP 是服务器端应用,所以它拥有对服务器端资源的访问权限,诸如 Servlet、JavaBean、EJB 和数据库。

2. JSP 工作原理

JSP 文件是在一个普通的静态 HTML 文件中添加了一些 Java 代码,JSP 文件的扩展名为. jsp。当第一次访问 JSP 页面的时候,这个文件首先会被 JSP 容器翻译为一个 Java 源文件,其实就是一个 Servlet,并进行编译生成相对应的字节码文件. class,然后像其他 Servlet 一样,由 Servlet 容器来处理。Servlet 容器装载这个类,处理来自客户的请求,并把结果返回给客户。这个过程如图 7-4 所示。

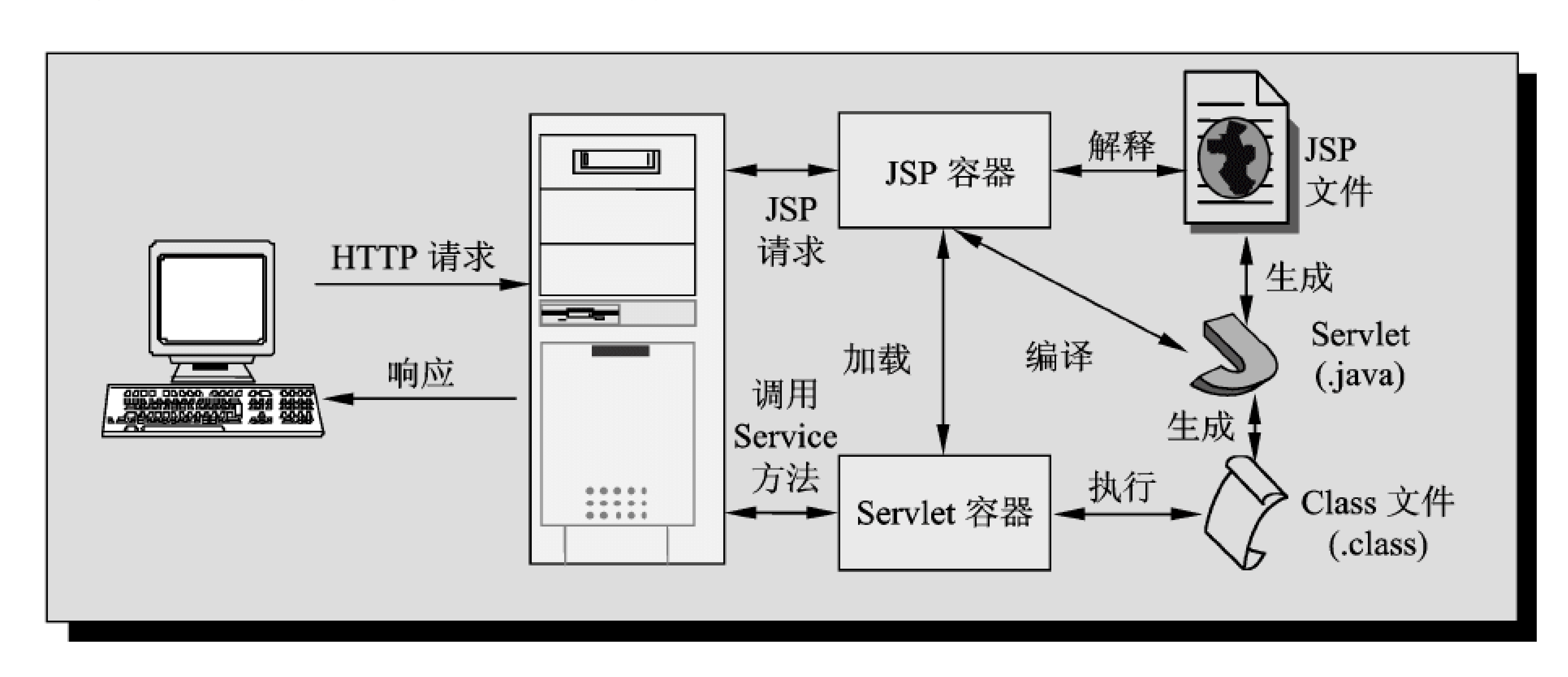


图 7-4 JSP 页面的执行过程

以后这个页面再被访问时,只要该文件没有发生过更改,JSP 容器就直接调用已经装载的 Servlet。如果已经做过修改,就会再次执行以上过程,翻译、编译并装载。因为首次访问的时候要执行一系列上面的过程,所以第一次访问某 JSP 页面时速度会较慢,但在以后运行时速度就会非常快。

Java2 企业版(J2EE)定义了几个容器:JSP 容器、Servlet 容器和 EJB 容器(企业级 JavaBean)。J2EE 规范定义的容器用于为企业级组件在其生命周期内或活动期内提供运行环境。J2EE 容器管理组件的生命周期并向组件提供不同的服务。此外,它们还协调组件与更大的运行环境之间的交互。

3. JSP 页面的生命周期

JSP 页面在被编译为 Servlet 并加载到 Servlet 容器后,Servlet 容器使用 3 个方法控制



其生命周期。这 3 个方法为 jspInit( )、jspSevice( )和 jspDestroy( )。这些方法是根据 JSP 页面的状态由 JSP 容器调用的。

在 javax. servlet. jsp 包中定义了一个 JspPage 接口，该接口定义了 jspInit( )与 jspDestroy( )两个方法。jspInit( )与 jspDestroy( )方法分别用于完成初始化和释放资源的操作。针对 HTTP 通信协议，javax. sevler. jsp 包定义了一个 HttpJspPage 接口。该接口只定义了一个 jspService( )方法，该方法由 JSP 容器调用，以响应浏览器端的 HTTP 请求。

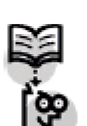
一般把 jspInit( )方法、jspService( )以及 jspDestroy( ) 3 个方法称为 JSP 生命周期方法。当一个 JSP 页面被请求调进时，由 JSP 容器把该 JSP 页面转换成一个 Servlet。在转换成功后，JSP 容器将调用 jspInit( )方法，创建 Servlet 的一个实例。

jspInit( )方法在 Servlet 生命周期只执行一次，然后将调用 jspService( )方法处理来自客户端的请求。当同时有多个客户请求时，JSP 容器将创建该 Servlet 的多个线程响应，这样每个客户请求对应一个 Servlet 线程，以多线程方式执行提高了系统的并发性。由于 Servlet 始终驻留在内存中，所以响应速度非常快。如果在 JSP 页面被转换成 Servlet 后，该 JSP 页面又被修改了，则 JSP 容器会重新编译该 JSP 页面，并用新生成的 Servlet 取代内存中旧的 Servlet。当不再需要一个 Servlet 时，jspDestroy( )方法将被调用，以释放该 Servlet 实例占用的系统资源。

7.3.4　经验总结

在 Web 程序的开发过程中，代码的重用是很重要的，很多代码都是相似的，尤其在 J2EE 的开发中，类、包的使用更是将这种重用发挥了出来，所以在开发的过程中要积累自己的包或者类库，开发出来的类要尽量地重用，这样不仅会提高工作效率，而且也是财富积累，这将大大加快日后的开发进度。

7.4　网站的安全性设计


知识储备

Web 应用程序的安全是系统开发人员必须考虑的重要因素之一，因为这涉及网站的建设者、网站的访问者等诸多安全问题，如果处理得不好，可能会给系统的访问者和管理者带来严重的问题。同时 Web 应用程序的安全解决方案不仅是技术问题，还涉及管理等多个方面的问题，尤其是服务器的配置，比如必须坚持服务器配置权限最小化原则等。

在此仅从 Web 应用程序的设计上来考虑安全性要求，对 Web 应用程序在开发过程中程序自身安全性进行设计，但由于有时并没有特意的引起重视，在每一个细节的处理时未注意网站的安全性，结果可能导致一些安全漏洞。

作为一名优秀的程序员，应充分考虑系统的安全性，这是程序员应具备的职业道德，也是系统经得起考验的基础。

1. MD5 数据加密技术

MD5 的全称是 Message-Digest Algorithm 5，用来对数据进行加密，用它对用户的登



录密码进行加密处理后，存到数据库中，这样可以保证私人密码的安全性。MD5 是目前比较流行和安全的加密方法。

没有经过 MD5 加密的密码直接显示在数据库表中，如果被黑客下载数据，查出数据库中的密码，或者内部开发人员通过数据库查出用户的密码，都可能会对用户的信息安全造成很大的影响。如果数据库表中存储 MD5 加密后的密码，在数据库中看到的是一连串经过加密的字符串，不能看到真正的密码，这样就能更好地保护网站的安全。

当用户登录的时候，系统把用户输入的密码计算成 MD5 值，然后再去和保存在文件系统中的 MD5 值进行比较，进而确定输入的密码是否正确。通过这样的步骤，系统在并不知道用户密码的明码的情况下就可以确定用户登录系统的合法性。这不但可以避免用户的密码被具有系统管理员权限的用户知道，而且还在一定程度上增加了密码被破解的难度。

使用 MD5 进行用户验证的过程如下：将客户端输入的验证信息进行 MD5 加密形成"密文 1"，发送到服务器端，服务器端从数据库读出验证信息的 MD5 值(密文 2)，然后"密文 1"与"密文 2"对比，若相等则认证成功，否则失败。

但是，如果"密文 1"在传输过程中被非法获取，非法用户即使不知道"密文 1"的内容，直接向服务器发送"密文 1"并请求验证，则验证可能成功，访问者的真实性无法保证。因此，需要对访问者的验证过程进行改进。在客户端请求验证的同时，通过 Ajax 技术异步向服务器申请一个临时的验证码，客户端将用户信息进行 n 次 MD5 混合运算后生成"密文 1"，附加验证码一起发送到服务器，服务器首先检查验证码是否与服务器端一致，若一致，到数据库中检索是否存在"密文 1"的用户，存在则成功，否则失败。验证码是改进后的验证关键，同时验证码还可以防止入侵者使用程序自动登录服务器，进行密码的暴力破解。因此，验证码技术上要求不能被复制，不能被扫描仪自动识别，不能随机生成，采用模糊的图片方式才能达到要求。

2. 验证码技术

普遍的客户端交互如留言本、会员注册等仅是按照要求输入内容，但网上有很多攻击软件，如注册机，可以通过浏览 Web，扫描表单，然后在系统上频繁注册，频繁发送不良信息，造成不良的影响，或者通过软件不断的尝试，盗取密码。通过使用验证码技术，使客户端输入的信息必须经过验证，从而可以解决这个问题。

所谓验证码，就是将一串随机产生的数字或符号，生成一幅图片，图片里加上一些干扰像素，由用户肉眼识别其中的验证码信息，输入表单提交网站验证，验证成功后才能使用某项功能。放在会员注册、留言本等所有客户端提交信息的页面，要提交信息，必须要输入正确的验证码，从而可以防止不法用户使用软件频繁注册，频繁发送不良信息等。

必须保证所有客户端交互部分都输入验证码，若提交信息时不输入验证码，或者故意输入错误的验证码，信息都不能提交，则说明验证码有效，若验证码输入正确则提交信息，说明验证码功能已完善。

3. 防 SQL 注入技术

SQL 注入式攻击是指在输入框或 URL 中输入 SQL 语句，绕过验证程序，非法获取用户的访问权，进行非法操作的入侵方式。防御 SQL 注入式攻击的方法常用两种，一种是使用数据库管理系统的存储过程，另一种是对输入的信息和 URL 请求信息中的敏感关键字过滤。

相当大一部分程序员在编写代码的时候，没有对用户输入数据的合法性进行判断，使应用程序存在安全隐患。新手最容易忽略的问题就是 SQL 注入漏洞的问题。用 NBSI 2.0 对



网上的网站扫描，就能发现部分网站存在 SQL 注入漏洞，访问者可以提交一段数据库查询代码，根据程序返回的结果，获得某些他想得知的数据。

当通过 URL、表单等向服务器提交信息时，利用一段防止 SQL 注入的过滤代码，即可防止出错信息暴露；或者通过转向，当系统出错时转到一个提示出错的页面等。同时服务器权限设置是一个非常重要的方面，由于涉及服务器的配置比较多，在此不作介绍。

对于文本型输入，如果要进行检查，就得根据字段本身的性质进行。例如年龄，可以限定此字段必须是数字，大小必须限定在一个范围之间，比如说 18～120 之间。对于用户名，应该建立一个集合，这个集合里存放有被允许的字符或被禁止的字符。

特别需要说明的是关于检查程序的问题。目前，程序对输入数据的检查是在前台通过客户端脚本完成的，这样攻击者很容易就能绕过检查程序。建议采用前、后台结合的方法，既可以保证效率，又可以提高安全性。

4．其他安全性设计

（1）URL 请求验证

URL 请求验证是防止用户非法请求的一种方法。非法请求是用户直接向服务器发送 URL 请求，在请求中传递一些非法参数，绕过系统的认证程序，以达到入侵者的非法目的。非法 URL 请求是入侵系统的常用手段。因此，进行 URL 请求的验证是提高系统安全性的一种有效方法。要实现 URL 的请求验证，同样采用验证码的方式，当用户登录网站时，通过 Ajax 技术异步向服务器发送申请验证码的请求，当打开网站的某一内容时，将内容打开请求信息与验证码一起发送到服务器，服务器进行验证码的核对，如果正确则显示，否则拒绝服务。

（2）双重验证

有些系统只在客户端进行验证，这是很不安全的。因为数据在传输过程中有可能被恶意篡改，服务器得到的将不是真实的数据，或者直接在 URL 中输验证请求，绕过客户端的验证程序，提交不安全的数据。因此，可以采用双重验证的方式，客户端的验证可以提高与用户的交互性，服务器端的验证保证数据的安全性。

（3）非法链接和非法复制

好多网站都使用框架结构，每个框架载入对应的页面，这些页面很容易被其他网站非法使用到自己的框架中。解决这一问题可以使用 URL 请求验证和图片打烙印相结合的方式。对于非法复制的问题，大多数网站采用禁止使用鼠标拖动选择、鼠标右击等功能，这些方法是为了防止不懂计算机的访问者，但稍有计算机知识的人还是很容易就能获取信息。因此，可以对一些字做成图片字典，当信息中包含这些字时，使用图片显示，用户非法复制后的内容将不是完整的内容（不过，编者认为，既然已经公布于网络上的信息，除非客户要求，否则没有必要这么做）。

（4）数据备份技术

当网站被黑客攻击或者其他原因丢失了数据，可以利用预先备份的数据恢复原来的数据，从而保证网站在一些人为的、自然的不可避免的条件下被损坏后的相对安全性。

采用数据库系统自动定时备份、自动定时删除几天以前的数据等，即可完成数据的备份功能。而图片、文件一般不能自动备份，需要定期手工对网站的图片、文件进行备份操作。备份间隔时间需要根据网站的更新频率来决定。

（5）参数篡改

参数篡改包括操纵 URL 字符串，以检索用户以其他方式得不到的信息。访问 Web 应



用的后端数据库是通过常常包含在 URL 中的 SQL 调用来进行的。恶意的访问者可以操纵 SQL 代码,以便将来有可能检索一份包含所有用户、口令、信用卡号的清单或者储存在数据库中的任何其他数据。

(6) 更改 Cookie

更改 Cookie 指的是修改存储在 Cookie 中的数据。网站常常将一些包括用户 ID、口令、账号等的 Cookie 存储到客户端的系统上。通过改变这些值,恶意的访问者就可以访问不属于他们的账户,攻击者也可以窃取他人的 Cookie 并访问他人的账户,达到不输入 ID 和口令或不进行其他验证就能访问系统的目的。

(7) 输入信息控制

输入信息检查包括通过控制由 CGI 脚本处理的 HTML 格式中的输入信息来运行系统命令。例如,使用 CGI 脚本向另一个用户发送信息的形式可以被攻击者控制来将服务器的口令文件邮寄给恶意的访问者或者删除系统上的所有文件。

(8) 缓冲区溢出

缓冲区溢出是恶意的访问者向服务器发送大量数据以使系统瘫痪的典型攻击手段,该系统包括存储这些数据的预置缓冲区。如果所收到的数据量大于缓冲区,则部分数据就会溢出到堆栈中。如果这些数据是代码,系统随后就会执行溢出到堆栈上的任何代码。Web 应用缓冲区溢出攻击的典型例子也涉及 HTML 文件。如果 HTML 文件上的一个字段中的数据足够的大,它就能创造一个缓冲区溢出条件。

(9) 直接访问浏览

直接访问浏览指直接访问应该需要验证的网页。没有正确配置的 Web 应用程序可以让恶意的访问者直接访问包括有敏感信息的 URL 或者使提供收费网页的公司丧失收入。

7.4.1 任务描述

进行“新闻类别网站”的安全性设计。

7.4.2 任务实现

任务 1　MD5 数据加密技术
任务 2　验证码技术
任务 3　防 SQL 注入技术
任务 4　其他安全性设计

任务 1：MD5 数据加密技术

该类主要是对网站用户的登录密码进行加密处理,存到数据库中,这样可以保证私人密码的安全性,是比较流行和安全的加密方法,此处的加密采用较为流行的 MD5 加密算法,加密过程为：将密码进行 MD5 加密后存入数据库,以后登录验证时将登录者输入的密码加密后与数据库中已存的加密密码对比,一致后则表示登录通过验证。具体操作步骤如下：

(1) 在 7.1 节任务 1 中建立的“com. news. util”包中新建类 MD5,保存为 MD5. java,大致写法如程序 7-27 所示的代码。



一般情况下，MD5 类不用自行编写，直接引用现成的就可以，在选用时要选用大公司或专业人员开发的，不要使用没有安全保障的。

【程序 7-27】 MD5. java

Example Source Code：

```
package com.news.util;                        //指定类所在的包
import java.lang.reflect.Array;
/*
 * Title:          @工作室
 * Description:    MD5 加密
 * Copyright:      Copyright (c) 2007
 * Company:        www.uc2008.net
 * author:         Skimob
 * version:        1.0

public class MD5
{
    static final int S11=7;
    static final int S12=12;
    static final int S13=17;
    static final int S14=22;
    static final int S21=5;
    static final int S22=9;
    static final int S23=14;
    static final int S24=20;
    static final int S31=4;
    static final int S32=11;
    static final int S33=16;
    static final int S34=23;
    static final int S41=6;
    static final int S42=10;
    static final int S43=15;
    static final int S44=21;
    static final byte PADDING[]={
        -128, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0
    };
    private long state[];
    private long count[];
    private byte buffer[];
    public String digestHexStr;
    private byte digest[];

    public MD5( )
    {
```



```
        state=new long[4];
        count=new long[2];
        buffer=new byte[64];
        digest=new byte[16];
        md5Init();
    }
    private void Decode(long al[], byte abyte0[], int i)
    {
        int j=0;
        for(int k=0; k<i; k+=4)
        {
            al[j]=b2iu(abyte0[k])|b2iu(abyte0[k+1])<<8|b2iu(abyte0[k+2])<<16|b2iu
            (abyte0[k+3])<<24;
            j++;
        }
    }
    private void Encode(byte abyte0[], long al[], int i)
    {
        int j=0;
        for(int k=0; k<i; k+=4)
        {
            abyte0[k]=(byte)(int)(al[j] & 255L);
            abyte0[k+1]=(byte)(int)(al[j]>>>8 & 255L);
            abyte0[k+2]=(byte)(int)(al[j]>>>16 & 255L);
            abyte0[k+3]=(byte)(int)(al[j]>>>24 & 255L);
            j++;
        }
    }
    private long F(long l, long l1, long l2)
    {
        return l & l1|~l & l2;
    }
    private long FF(long l, long l1, long l2, long l3, long l4, long l5, long l6)
    {
        l+=F(l1, l2, l3)+l4+l6;
        l=(int)l<<(int)l5|(int)l>>>(int)(32L-l5);
        l+=l1;
        return l;
    }
    private long G(long l, long l1, long l2)
    {
        return l & l2|l1 & ~l2;
    }
    private long GG(long l, long l1, long l2, long l3, long l4, long l5, long l6)
    {
        l+=G(l1, l2, l3)+l4+l6;
        l=(int)l<<(int)l5|(int)l>>>(int)(32L-l5);
        l+=l1;
        return l;
    }
    private long H(long l, long l1, long l2)
```



```
{
    return l ^ l1 ^ l2;
}
private long HH(long l, long l1, long l2, long l3, long l4, long l5, long l6)
{
    l+=H(l1, l2, l3)+l4+l6;
    l=(int)l<<(int)l5|(int)l>>>(int)(32L-l5);
    l+=l1;
    return l;
}
private long I(long l, long l1, long l2)
{
    return l1 ^ (l|~l2);
}
private long II(long l, long l1, long l2, long l3, long l4, long l5, long l6)
{
    l+=I(l1, l2, l3)+l4+l6;
    l=(int)l<<(int)l5|(int)l>>>(int)(32L-l5);
    l+=l1;
    return l;
}
public static long b2iu(byte byte0)
{
    return (long)(byte0>=0 ? byte0:  byte0 & 0xff);
}
public static String byteHEX(byte byte0)
{
    char ac[]={
        '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
        'A', 'B', 'C', 'D', 'E', 'F'
    };
    char ac1[]=new char[2];
    ac1[0]=ac[byte0>>>4 & 0xf];
    ac1[1]=ac[byte0 & 0xf];
    String s=new String(ac1);
    return s;
}
public String getMD5ofStr(String s)
{
    md5Init();
    md5Update(s.getBytes(), s.length());
    md5Final();
    digestHexStr="";
    for(int i=0; i<16; i++)
        digestHexStr+=byteHEX(digest[i]);
    return digestHexStr;
}
public static void main(String args[])
{
    MD5 md5=new MD5();
    if(Array.getLength(args)==0)
```



```
        {
            System.out.println("MD5 Test suite:");
            System.out.println("MD5(\"\"):"+md5.getMD5ofStr(""));
            System.out.println("MD5(\"a\"):"+md5.getMD5ofStr("a"));
            System.out.println("MD5(\"abc\"):"+md5.getMD5ofStr("abc"));
             System.out.println("MD5(\"message digest\"):"+md5.getMD5ofStr("message
             digest"));
             System.out.println("MD5(\"abcdefghijklmnopqrstuvwxyz\"):"+md5.getMD5ofStr("
             abcdefghijklmnopqrstuvwxyz"));
              System.out.println("MD5(\"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuv
              wxyz0123456789\"):"+md5.getMD5ofStr("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijk
              lmnopqrstuvwxyz0123456789"));
        } else
        {
            System.out.println("MD5("+args[0]+")="+md5.getMD5ofStr(args[0]));
        }
    }
    private void md5Final()
    {
        byte abyte0[]=new byte[8];
        Encode(abyte0, count, 8);
        int i=(int)(count[0]>>>3) & 0x3f;
        int j=i >=56 ?120-i:  56-i;
        md5Update(PADDING, j);
        md5Update(abyte0, 8);
        Encode(digest, state, 16);
    }
    private void md5Init()
    {
        count[0]=0L;
        count[1]=0L;
        state[0]=0x67452301L;
        state[1]=0xefcdab89L;
        state[2]=0x98badcfeL;
        state[3]=0x10325476L;
    }
    private void md5Memcpy(byte abyte0[], byte abyte1[], int i, int j, int k)
    {
        for(int l=0; l<k; l++)
            abyte0[i+l]=abyte1[j+l];
    }
    private void md5Transform(byte abyte0[])
    {
        long l=state[0];
        long l1=state[1];
        long l2=state[2];
        long l3=state[3];
        long al[]=new long[16];
        Decode(al, abyte0, 64);
        l=FF(l, l1, l2, l3, al[0], 7L, 0xd76aa478L);
        l3=FF(l3, l, l1, l2, al[1], 12L, 0xe8c7b756L);
```






```
l2=FF(l2, l3, l, l1, al[2], 17L, 0x242070dbL);
l1=FF(l1, l2, l3, l, al[3], 22L, 0xc1bdceeeL);
l=FF(l, l1, l2, l3, al[4], 7L, 0xf57c0fafL);
l3=FF(l3, l, l1, l2, al[5], 12L, 0x4787c62aL);
l2=FF(l2, l3, l, l1, al[6], 17L, 0xa8304613L);
l1=FF(l1, l2, l3, l, al[7], 22L, 0xfd469501L);
l=FF(l, l1, l2, l3, al[8], 7L, 0x698098d8L);
l3=FF(l3, l, l1, l2, al[9], 12L, 0x8b44f7afL);
l2=FF(l2, l3, l, l1, al[10], 17L, 0xffff5bb1L);
l1=FF(l1, l2, l3, l, al[11], 22L, 0x895cd7beL);
l=FF(l, l1, l2, l3, al[12], 7L, 0x6b901122L);
l3=FF(l3, l, l1, l2, al[13], 12L, 0xfd987193L);
l2=FF(l2, l3, l, l1, al[14], 17L, 0xa679438eL);
l1=FF(l1, l2, l3, l, al[15], 22L, 0x49b40821L);
l=GG(l, l1, l2, l3, al[1], 5L, 0xf61e2562L);
l3=GG(l3, l, l1, l2, al[6], 9L, 0xc040b340L);
l2=GG(l2, l3, l, l1, al[11], 14L, 0x265e5a51L);
l1=GG(l1, l2, l3, l, al[0], 20L, 0xe9b6c7aaL);
l=GG(l, l1, l2, l3, al[5], 5L, 0xd62f105dL);
l3=GG(l3, l, l1, l2, al[10], 9L, 0x2441453L);
l2=GG(l2, l3, l, l1, al[15], 14L, 0xd8a1e681L);
l1=GG(l1, l2, l3, l, al[4], 20L, 0xe7d3fbc8L);
l=GG(l, l1, l2, l3, al[9], 5L, 0x21e1cde6L);
l3=GG(l3, l, l1, l2, al[14], 9L, 0xc33707d6L);
l2=GG(l2, l3, l, l1, al[3], 14L, 0xf4d50d87L);
l1=GG(l1, l2, l3, l, al[8], 20L, 0x455a14edL);
l=GG(l, l1, l2, l3, al[13], 5L, 0xa9e3e905L);
l3=GG(l3, l, l1, l2, al[2], 9L, 0xfcefa3f8L);
l2=GG(l2, l3, l, l1, al[7], 14L, 0x676f02d9L);
l1=GG(l1, l2, l3, l, al[12], 20L, 0x8d2a4c8aL);
l=HH(l, l1, l2, l3, al[5], 4L, 0xfffa3942L);
l3=HH(l3, l, l1, l2, al[8], 11L, 0x8771f681L);
l2=HH(l2, l3, l, l1, al[11], 16L, 0x6d9d6122L);
l1=HH(l1, l2, l3, l, al[14], 23L, 0xfde5380cL);
l=HH(l, l1, l2, l3, al[1], 4L, 0xa4beea44L);
l3=HH(l3, l, l1, l2, al[4], 11L, 0x4bdecfa9L);
l2=HH(l2, l3, l, l1, al[7], 16L, 0xf6bb4b60L);
l1=HH(l1, l2, l3, l, al[10], 23L, 0xbebfbc70L);
l=HH(l, l1, l2, l3, al[13], 4L, 0x289b7ec6L);
l3=HH(l3, l, l1, l2, al[0], 11L, 0xeaa127faL);
l2=HH(l2, l3, l, l1, al[3], 16L, 0xd4ef3085L);
l1=HH(l1, l2, l3, l, al[6], 23L, 0x4881d05L);
l=HH(l, l1, l2, l3, al[9], 4L, 0xd9d4d039L);
l3=HH(l3, l, l1, l2, al[12], 11L, 0xe6db99e5L);
l2=HH(l2, l3, l, l1, al[15], 16L, 0x1fa27cf8L);
l1=HH(l1, l2, l3, l, al[2], 23L, 0xc4ac5665L);
l=II(l, l1, l2, l3, al[0], 6L, 0xf4292244L);
l3=II(l3, l, l1, l2, al[7], 10L, 0x432aff97L);
l2=II(l2, l3, l, l1, al[14], 15L, 0xab9423a7L);
l1=II(l1, l2, l3, l, al[5], 21L, 0xfc93a039L);
l=II(l, l1, l2, l3, al[12], 6L, 0x655b59c3L);
```



```
            l3=II(l3, l, l1, l2, al[3], 10L, 0x8f0ccc92L);
            l2=II(l2, l3, l, l1, al[10], 15L, 0xffeff47dL);
            l1=II(l1, l2, l3, l, al[1], 21L, 0x85845dd1L);
            l=II(l, l1, l2, l3, al[8], 6L, 0x6fa87e4fL);
            l3=II(l3, l, l1, l2, al[15], 10L, 0xfe2ce6e0L);
            l2=II(l2, l3, l, l1, al[6], 15L, 0xa3014314L);
            l1=II(l1, l2, l3, l, al[13], 21L, 0x4e0811a1L);
            l=II(l, l1, l2, l3, al[4], 6L, 0xf7537e82L);
            l3=II(l3, l, l1, l2, al[11], 10L, 0xbd3af235L);
            l2=II(l2, l3, l, l1, al[2], 15L, 0x2ad7d2bbL);
            l1=II(l1, l2, l3, l, al[9], 21L, 0xeb86d391L);
            state[0]+=l;
            state[1]+=l1;
            state[2]+=l2;
            state[3]+=l3;
        }
        private void md5Update(byte abyte0[], int i)
        {
            byte abyte1[]=new byte[64];
            int k=(int)(count[0]>>>3) & 0x3f;
            if((count[0]+=i<<3)<(long)(i<<3))
                count[1]++;
            count[1]+=i >>>29;
            int l=64-k;
            int j;
            if(i >=l)
            {
                md5Memcpy(buffer, abyte0, k, 0, l);
                md5Transform(buffer);
                for(j=l; j+63<i; j+=64)
                {
                    md5Memcpy(abyte1, abyte0, 0, j, 64);
                    md5Transform(abyte1);
                }
                k=0;
            } else
            {
                j=0;
            }
            md5Memcpy(buffer, abyte0, k, j, i-j);
        }
    }
```

（2）在进行加密处理时，首先要新建 MD5 类，使其实例化，而后才能调用 getMD5ofStr( ) 方法进行编码，才能起到加密的作用，也可进行双重编码，即对编码所得的再次进行 MD5 加密编码处理，从而增强安全性。具体调用方法如程序 7-28 代码所示。

【程序 7-28】 MD5 调用示例

Example Source Code：

```
//实例化 MD5 类
```



```
MD5 md5=new MD5();
...
//调用 getMD5ofStr()方法进行编码
...
md5.getMD5ofStr(Pwd)
...
```

任务 2：验证码技术

为了防止恶意访问者利用机器人在网站上自动注册、登录、灌水，"新闻类别网站"采用了验证码技术。所谓验证码，就是将一串随机产生的数字或符号，生成一幅图片，图片里加上一些干扰像素(防止 OCR)，由访问者肉眼识别其中的验证码信息，输入表单提交网站验证，验证成功后才能使用某项功能。具体操作步骤如下：

(1) 在"com. news. util"包中新建基本类 VerifyCode，保存为 VerifyCode. java，大致写法如程序 7-29 代码所示。

【程序 7-29】 基本类 VerifyCode. java

Example Source Code：

```
package com.news.util;                                    //指定类所在的包
import java.awt.*;                                         //导入类
import java.awt.image.*;
import java.util.*;

//定义类
public class VerifyCode {
    static Random r=new Random();
    static String ssource="ABCDEFGHIJKLMNOPQRSTUVWXYZ"+"abcdefghijklmnopqrstuvwxyz"+"
        0123456789";
    static char[] src=ssource.toCharArray();

    //产生随机字符串
    private static String randString (int length){
        char[] buf=new char[length];
        int rnd;
        for(int i=0;i<length;i++){
            rnd=Math.abs(r.nextInt())%src.length;

            buf[i]=src[rnd];
        }
        return new String(buf);
    }

    //调用该方法，产生随机字符串，
    //参数 i：为字符串的长度
    public String runVerifyCode(int i){
        String VerifyCode=randString(i);
        return VerifyCode;
    }

    //给定范围获得随机颜色
```



```
public Color getRandColor(int fc,int bc)
{
   Random random=new Random();
   if(fc> 255) fc=255;
   if(bc> 255) bc=255;
   int r=fc+random.nextInt(bc-fc);
   int g=fc+random.nextInt(bc-fc);
   int b=fc+random.nextInt(bc-fc);
   return new Color(r,g,b);
   }
  //调用该方法将得到的验证码生成图像
  //sCode:传递验证码 w:图像宽度 h:图像高度
  public BufferedImage CreateImage(String sCode)
  {
      try{
          //字符的字体
        Font CodeFont=new Font("Arial Black",Font.PLAIN,14);
        int iLength=sCode.length();                    //得到验证码长度
        int width=18*iLength, height=20;               //图像宽度与高度
        int CharWidth=(int)(width-24)/iLength;         //字符距左边宽度
        int CharHeight=16;                             //字符距上边高度

        // 在内存中创建图像
        BufferedImage image=new BufferedImage(width,height,BufferedImage.TYPE_INT_
        RGB);

        // 获取图形上下文
        Graphics g=image.getGraphics();

        //生成随机类
        Random random=new Random();

        // 设定背景色
        g.setColor(getRandColor(200,240));
        g.fillRect(0, 0, width, height);

        //设定字体
        g.setFont(CodeFont);

        //画随机颜色的边框
        g.setColor(getRandColor(10,50));
        g.drawRect(0,0,width-1,height-1);

        // 随机产生 155 条干扰线,使图像中的认证码不易被其他程序探测到
        g.setColor(getRandColor(160,200));
        for (int i=0;i<155;i++)
        {
              int x=random.nextInt(width);
              int y=random.nextInt(height);
              int xl=random.nextInt(12);
              int yl=random.nextInt(12);
```



```
                g.drawLine(x,y,x+xl,y+yl);
            }
            for (int i=0;i<iLength;i++)
            {
                String rand=sCode.substring(i,i+1);
                //将认证码显示到图像中
                g.setColor(new Color(20+random.nextInt(60),20+random.nextInt(120),20+
                random.nextInt(180)));
                g.drawString(rand,CharWidth*i+14,CharHeight);
            }
            //图像生效
            g.dispose();
            return image;
        }catch(Exception e){
            //e.printStackTrace();
            //System.out.println(e.getMessage());
            }
        return null;
    }
}
```

（2）在 Eclipse 中新建一个 JSP 文件，作为验证码输出的 JSP 页面，保存到项目中，文件名为 VerifyCode.jsp，具体步骤可参看 5.6 节的任务 2，具体实现参照程序 7-30 所示代码。

【程序 7-30】 VerifyCode.jsp

Example Source Code：

```
<meta http-equiv="Content-Type" content="text/html; charset=GBK">
<%@page contentType="text/html; charset=gb2312" language="java"
    import="java.awt.*,java.awt.image.*"%>
<%@page import="java.util.*,javax.imageio.*,java.io.*"%>
<%@page import="com.news.util.VerifyCode"%>
<%!String sVerifyCode;          //验证码字符串  %>
<%    //设置页面不缓存
    response.setHeader("Pragma","No-cache");
    response.setHeader("Cache-Control","no-cache");
    response.setDateHeader("Expires", 0);
    response.reset();

    try{
        //将认证码存入 SESSION
        //调用 runVerifyCode(int i),把 i 改成所要的验证码位数
        VerifyCode VC=new VerifyCode();
        session.setAttribute("VerifyCode",VC.runVerifyCode(4));

        //输出图像到页面
        sVerifyCode=(String)session.getAttribute("VerifyCode");
        OutputStream outs=response.getOutputStream();
        ImageIO.write(VC.CreateImage(sVerifyCode),"JPEG",outs);
    }catch(Exception e){
        return;
```



```
    }
%>
```

任务 3：防止 SQL 注入技术

SQL 注入是一种攻击方式，在这种攻击方式中，恶意代码被插入到字符串中，然后将该字符串传递到 SQL Server 的实例以进行分析和执行。任何构成 SQL 语句的过程都应进行注入漏洞检查，因为 SQL Server 将执行其接收到的所有语法有效的查询。一个有经验的、坚定的攻击者甚至可以操作参数化数据。

SQL 注入的主要形式包括直接将代码插入到与 SQL 命令串联在一起并使其得以执行的用户输入变量。一种间接的攻击会将恶意代码注入在表中存储或作为元数据存储的字符串中。在存储的字符串随后串连到一个动态 SQL 命令中时，将执行该恶意代码。

注入过程的工作方式是提前终止文本字符串，然后追加一个新的命令。由于插入的命令可能在执行前追加其他字符串，因此攻击者将用注释标记“--”来终止注入的字符串。执行时，此后的文本将被忽略。具体操作步骤如下：

（1）在本章 7.1 节任务 1 中建立的“com.news.util”包中的 Format 类下新建 CheckReplace(String s)方法，具体实现参照程序 7-31 所示代码。

【程序 7-31】 CheckReplace 函数

Example Source Code：

```
/****************************************************************************
* 函数名：CheckReplace
* 作用：转化 SQL 特殊字符串
* 参数：s：字符串型，待转化的字符
* 返回值：转化以后的字符串
* 调用：String s2=CheckReplace(s1);
****************************************************************************/
public String CheckReplace(String s) {
    try {
        if (s==null || s.equals(""))
            return "";
        else {
            StringBuffer stringbuffer=new StringBuffer();
            for (int i=0; i<s.length(); i++) {
                char c=s.charAt(i);
                switch (c) {
                    case 34: // '"'
                        stringbuffer.append("&quot;");
                        break;
                    case 39: // '\''
                        stringbuffer.append("&#039;");
                        break;
                    case 124: // '|'
                        stringbuffer.append("");
                        break;
                    case '&':
                        stringbuffer.append("&amp;");
                        break;
```



```
                    case '<':
                        stringbuffer.append("&lt;");
                        break;
                    case '>':
                        stringbuffer.append("&gt;");
                        break;
                    default:
                        stringbuffer.append(c);
                        break;
                }
            }
            return stringbuffer.toString().trim(); // 返回转化以后的字符串
        }
    } catch (Exception e) {
        return "";
    }
}
```

（2）在"com.news.util"包中 Format 类下，除上述 CheckReplace(String s)方法外，还建有 toSql(String str)方法，具体实现参照程序 7-31 所示代码，它们都是用于过滤 SQL 的特殊字符的。

关于 SQL 注入的防范还有很多的方法，在此使用的只是诸多方法中的最基本的方法，而且只是一些特殊字符的过滤，并不涉及其他更高级的处理，想要更深层次了解 SQL 注入的防范的读者可自己查找相关的材料。

任务 4：其他安全性设计

一个系统中关于安全性的设计是很多的，除了上述的安全考虑外，本系统还设计了网站用户登录验证的安全考虑。具体思路为：在用户登录时形成一个 Session 变量，并对其设置有效时间，在这个时间内该 Session 变量有效，而且用户打开其他需要验证用户是否登录的页面时，只需读取该 Session 变量进行检查就可以了。具体操作步骤如下：

（1）在 Eclipse 中新建一个 JSP 文件，作为验证码输出的 JSP 页面，保存到项目中，文件名为 Session.jsp，具体步骤可参看 5.6 节的任务 2，具体实现参照程序 7-32 所示代码。

【程序 7-32】 Session.jsp

Example Source Code:

```
<%@page import="com.news.*"%>
<%
    response.setHeader("Pragma","No-cache");
    response.setHeader("Cache-Control","no-cache");
    response.setDateHeader("Expires", 0);
    request.setCharacterEncoding("GBK");

    String isLogin=(String)session.getAttribute("Login");
    if(isLogin==null || !isLogin.equals("Yes"))
    {
    out.println(new String("<SCRIPT LANGUAGE='JavaScript'>alert('对不起,您没有登录系统,请登
    录!');location.href='Login.jsp';</SCRIPT>".getBytes("ISO-8859-1"), "GBK"));
    }
%>
```



（2）在需要验证的JSP页面中，利用引入Session.jsp，就可以起到验证的作用，具体引用参照如下格式：

```
<%@include file="Session.jsp"%>
```

7.4.3 相关知识拓展

1. MD5

MD5，在20世纪90年代初由MIT的计算机科学实验室和RSA Data Security inc发明，经MD2、MD3和MD4发展而来。

Message-Digest泛指字节串（Message）的Hash变换，它把一个任意长度的字节串变换成一个128bit的大整数，并且它是一个不可逆的字符串变换算法，也就是说，即使能看到源程序和算法描述，也无法将一个MD5的值变换回原始的字符串。

MD5的典型应用是对一段Message（字节串）产生fingerprint（指纹），以防止被“篡改”。例如，将一段话写在一个叫readme.txt的文件中，对这个readme.txt产生一个MD5的值并记录在案，然后将这个文件传播给他人，若他人修改了这个文件中的任何内容，再对这个文件重新计算MD5时就会被发现；如果再有一个第三方的认证机构，用MD5还可以防止文件作者的“抵赖”，这就是所谓的数字签名应用。

MD5还广泛应用于加密与解密技术上，在很多操作系统中，用户的密码是以MD5值（或类似的其他算法）的方式保存的，用户登录的时候，系统把用户输入的密码计算成MD5的值，然后再和系统中保存的MD5值进行比较，而系统并不“知道”用户的实际密码是什么。

2. 防御和检查SQL注入的手段

（1）使用参数化的过滤性语句。要防御SQL注入，浏览者提交的信息就绝对不能直接被嵌入到SQL语句中。恰恰相反，来自浏览器端的输入必须进行过滤，或者使用参数化的语句。参数化的语句要使用参数，而不是将输入信息直接嵌入到SQL语句中。在多数情况中，这种方法可以修正SQL语句。下面是一个使用Java和JDBC API的例子：

```
PreparedStatement prep=conn.prepareStatement("SELECT * FROM USERS WHERE PASSWORD=?");
prep.setString(1, pwd);
```

总体上讲，有两种方法可以保证应用程序不易受到SQL注入的攻击，一是使用代码复查，二是强迫使用参数化语句。强迫使用参数化语句意味着嵌入输入信息的SQL语句在运行时将被拒绝。

（2）避免使用解释程序，因为这正是黑客们借以执行非法命令的手段。

（3）避免出现一些详细的错误消息，因为黑客们可以利用这些消息。要使用一种标准的输入确认机制来验证所有的输入数据的长度、类型、语句、企业规则等。

（4）使用专业的漏洞扫描工具。攻击者们目前正在自动搜索攻击目标并实施攻击，其攻击技术甚至可以轻易地被应用于其他的Web架构中的漏洞。企业应当投资于一些专业的漏洞扫描工具，如Acunetix的Web漏洞扫描程序等。漏洞扫描程序专门查找网站上的

SQL 注入式漏洞。最新的漏洞扫描程序可以查找最新发现的漏洞。

（5）企业要在 Web 应用程序开发过程的所有阶段实施代码的安全检查。首先，要在部署 Web 应用之前实施安全测试，这种措施的意义比以前更大、更深远。企业还应当在部署之后用漏洞扫描工具和站点监视工具对网站进行测试。

7.4.4　经验总结

Web 安全性测试可以从部署、输入验证、身份验证、授权、配置管理、敏感数据、会话管理、加密、参数操作、异常管理、审核和日志记录等几个方面入手。

1. SESSION 安全问题（关闭页面时自动清除 SESSION）

如果直接关闭浏览器，不会直接触发 SESION_ONEND 事件，因此为了安全的需要，就应调用页面关闭触发的事件 onUnload，利用这个事件来执行一个函数。在函数中调用 session.abandon 事件，真正实现没有漏洞的 SESSION 清除。但如果只是单独利用 session.abandon，将导致直接关闭页面时 SESSION 还存在；如果只是利用服务器自动引发的 SESSION_TIMEOUT 事件，将会导致在设置时间没有结束的一段时间内 SESSION 还存在，这两种方式都存在安全漏洞。

2. 如何防止 IE 缓存 jsp 文件

（1）使用 Java 提供的方法

```
<%
response.setHeader("Pragma","No-cache");
response.setHeader("Cache-Control","no-cache");
response.setDateHeader("Expires", 0);
%>
```

（2）使用 HTML 标记

```
<HEAD>
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache">
<META HTTP-EQUIV="Expires" CONTENT="0">
</HEAD>
```

许多人认为数据备份就只是数据库的备份，其实还有动态变化的图片、文件等也需要备份。

7.5　实　　训

根据本章所述内容，完成“新闻类别网站”服务器端功能的编程任务。

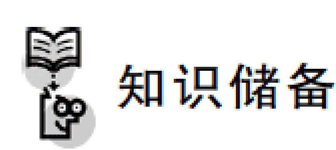


第8章　存取数据库

存取数据库，就是对数据库进行添加、修改、删除和查询等操作，是整个系统的最底层。

本章在第7章业务层的基础上，依据进行数据存取的先后顺序，从使用JDBC连接数据库，到操作数据库的常用命令，再通过实现系统中新闻的添加、修改、删除和查询等操作的实例，尤其将管理员的登录模块单独拿出来讲解，全面而详细地讲解了对数据库的存取操作。

8.1　连接数据库



知识储备

在Java语言中，JDBC(Java DataBase Connection)是应用程序与数据库沟通的桥梁，即Java语言通过JDBC技术访问数据库。JDBC是一种“开放”的方案，它为数据库应用开发人员、数据库前台工具开发人员提供了一种标准的应用程序设计接口，使开发人员可以用纯Java语言编写完整的数据库应用程序。JDBC提供两种API，分别是面向开发人员的API和面向底层的JDBC驱动程序API，底层主要通过直接的JDBC驱动和JDBC—ODBC桥驱动实现与数据库的连接。

1. Java应用程序访问数据库的过程

(1) 装载数据库驱动程序。

(2) 通过JDBC建立数据库连接。

(3) 访问数据库，执行SQL语句。

(4) 断开数据库连接。

JDBC作为一种数据库访问技术，具有简单易用的优点。但使用这种模式进行Web应用程序开发，存在很多问题：首先，每一次Web请求都要建立一次数据库连接。建立连接是一个费时的活动，每次都得花费0.05～1s的时间，而且系统还要分配内存资源。这个时间对于一次或几次数据库操作，或许感觉不出系统有多大的开销，可是对于现在的Web应用，尤其是大型电子商务网站，同时有几百人甚至几千人在线是很正常的事，在这种情况下，频繁地进行数据库连接操作势必占用很多的系统资源，网站的响应速度必定下降，严重的甚至会造成服务器的崩溃。这也是制约某些电子商务网站发展的技术瓶颈问题。其次，对于每一次数据库连接，使用完后都得断开，否则，如果程序出现异常而未能关闭，将会导致数据库系统中的内存泄漏，最终将不得不重启数据库。再次，这种开发不能控制被创建的连接对



象数，系统资源会被毫无顾忌地分配出去，如连接过多，也可能导致内存泄漏，服务器崩溃。

2. 数据库连接池(connection pool)的工作原理

(1) 基本概念及原理

由上面的分析可以看出，使用 JDBC 技术访问数据库存在缺陷的根源在于对数据库连接资源的低效管理。

对于共享资源，有一个很著名的设计模式：资源池(Resource Pool)。该模式正是为了解决资源的频繁分配、释放所造成的问题。为解决上述问题，可以采用数据库连接池技术。数据库连接池的基本思想就是为数据库连接建立一个“缓冲池”。预先在缓冲池中放入一定数量的连接，当需要建立数据库连接时，只需从“缓冲池”中取出一个，使用完毕之后再放回去。可以通过设定连接池最大连接数来防止系统无尽地与数据库连接。更重要的是可以通过连接池的管理机制监视数据库的连接数量、使用情况，为系统开发、测试及性能调整提供依据。

(2) 服务器自带的连接池

JDBC 的 API 中没有提供连接池的方法。一些大型的 Web 应用服务器如 BEA 的 WebLogic 和 IBM 的 WebSphere 等提供了连接池机制，但必须使用其第三方的专用类方法来支持连接池的用法。

3. 连接池关键问题分析

(1) 并发问题

为了使连接管理服务具有最大的通用性，必须考虑多线程环境，即并发问题。Java 语言提供了对并发管理的支持，使用 synchronized 关键字即可确保线程是同步的。使用方法是直接在类方法前面加上 synchronized 关键字，如：

```
public synchronized Connection getConnection()
```

(2) 多数据库服务器和多用户

对于大型企业级应用，常常需要同时连接不同的数据库(如连接 Oracle 和 Sybase)。如何连接不同的数据库呢？可以采用的策略是：设计一个符合单例模式的连接池管理类，在连接池管理类的唯一实例被创建时读取一个资源文件，其中资源文件中存放着多个数据库的 url 地址(<poolName. url>)、用户名(<poolName. user>)、密码(<poolName. password>)等信息。如“tx. url＝172. 21. 15. 123：5000/tx_it，tx. user＝yang，tx. password＝yang321”。根据资源文件提供的信息，创建多个连接池类的实例，每一个实例都是一个特定数据库的连接池。连接池管理类实例为每个连接池实例取一个名字，通过不同的名字来管理不同的连接池。

对于同一个数据库有多个用户使用不同的名称和密码访问的情况，也可以通过资源文件处理，在资源文件中设置多个具有相同 URL 地址，但具有不同用户名和密码的数据库连接信息。

4. 事务处理

事务具有原子性，要求对数据库的操作符合“ALL—ALL—NOTHING”原则，即对于一组 SQL 语句要么全部执行，要么全部不执行。

在 Java 语言中，Connection 类本身提供了对事务的支持，可以通过设置 Connection 的 AutoCommit 属性为 false，然后显式的调用 commit 或 rollback 方法来实现。但要高效地进



行 Connection 复用，就必须提供相应的事务支持机制。可采用每一个事务独占一个连接来实现，这种方法可以大大降低事务管理的复杂性。

5. 连接池的分配与释放

连接池的分配与释放对系统的性能有很大的影响。合理的分配与释放，可以提高连接的复用度，从而降低建立新连接的开销，同时还可以加快用户的访问速度。

对于连接的管理可使用空闲池，即把已经创建但尚未分配出去的连接按创建时间存放到一个空闲池中。每当访问者请求一个连接时，系统首先检查空闲池内有没有空闲连接。如果有就把建立时间最长(通过容器的顺序存放实现)的那个连接分配给他(实际是先做连接是否有效的判断，如果可用就分配给访问者，如不可用就把这个连接从空闲池删掉，重新检测空闲池是否还有连接)；如果没有，则检查当前所开连接池是否达到连接池所允许的最大连接数(maxConn)，如果没有达到，就新建一个连接，如果已经达到，就等待一定的时间(timeout)。如果在等待的时间内有连接被释放出来，就可以把这个连接分配给等待的用户；如果等待时间超过预定时间 timeout，则返回空值(null)。系统对已经分配出去且正在使用的连接只进行计数，当使用完后再返还给空闲池。对于空闲连接的状态，可开辟专门的线程定时检测，这样会花费一定的系统开销，但可以保证较快的响应速度。也可采取不开辟专门线程，只是在分配前检测的方法。

6. 连接池的配置与维护

(1) 连接池中的连接数配置

系统可以设置最小连接数(minConn)和最大连接数(maxConn)来控制连接池中的连接。最小连接数是系统启动时连接池所创建的连接数。如果创建过多，则系统启动就慢，但创建后系统的响应速度会很快；如果创建过少，则系统启动的很快，响应起来却慢。这样，可以在开发时，设置较小的最小连接数，开发起来会快，而在系统实际使用时设置较大的，访问速度会快些。最大连接数是连接池中允许连接的最大数目，具体设置多少，要看系统的访问量，可通过反复测试，找到最佳点。

(2) 确保连接池中的最小连接数策略

可以采用动态和静态两种策略。动态即每隔一定时间就对连接池进行检测，如果发现连接数量小于最小连接数，则补充相应数量的新连接，以保证连接池的正常运转。静态是发现空闲连接不够时再去检查。

7. "新闻类别网站"采用的数据操作组件

应用程序访问数据库通过自定义的数据访问类(DBConnect)来实现，数据库访问类通过自定义的数据库连接池类(DBConnectionManager)来实现数据访问。通过设置\WEB—INF\classes\目录下的 db. properties 配置文件中的数据库配置部分，配置数据库连接。

这种模式中，调用者进行数据操作的时序如图 8-1 所示。

各类之间的相互调用和协作如图 8-2 所示。

说明：

(1) 创建 DBConnect( )对象。

(2) 用 getInstance( )方法创建连接实例。

(3) 用 createPools( )方法创建连接池实例。

(4) 用 loadDrivers( )方法装载和注册所有 JDBC 驱动程序。



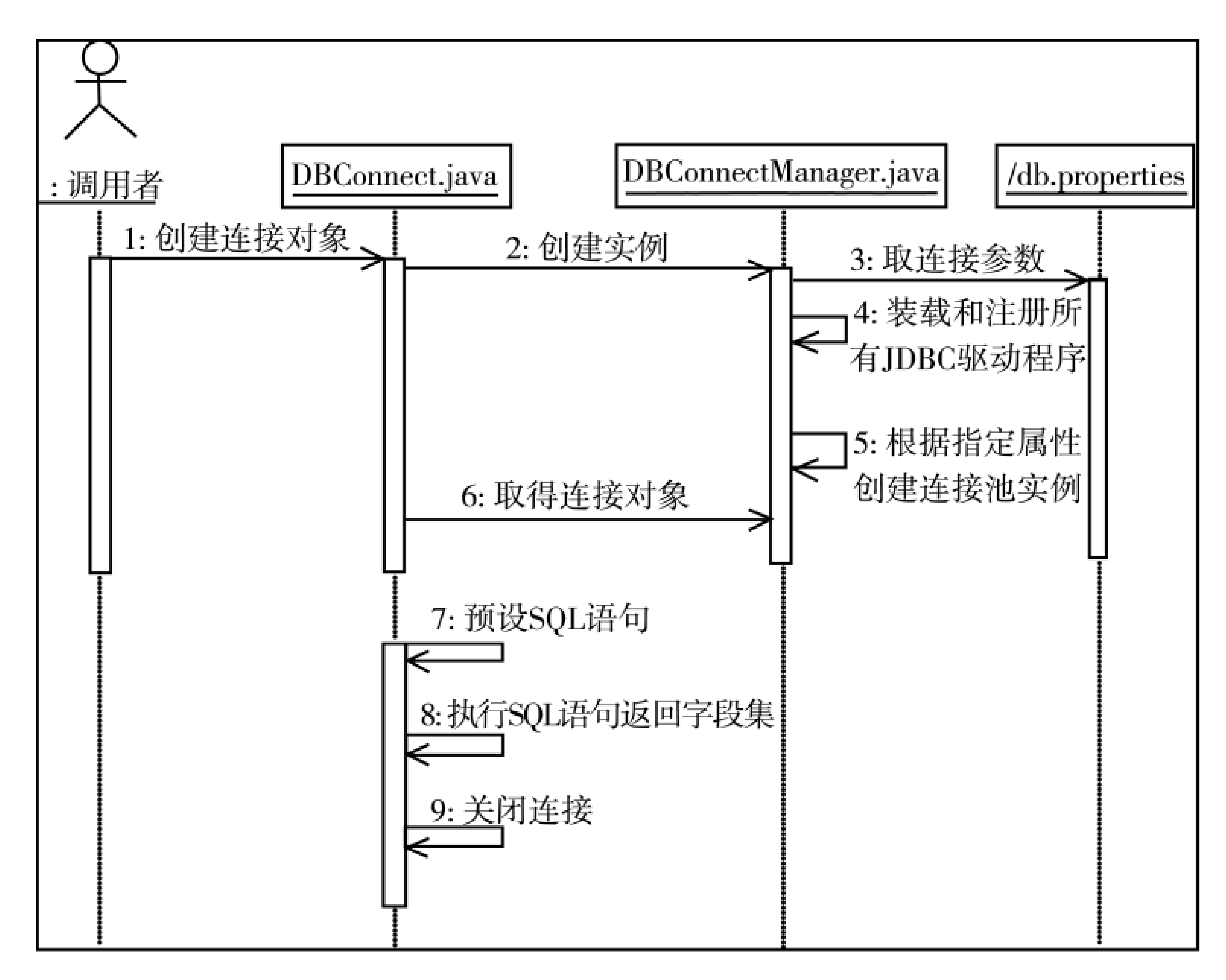


图 8-1　时序图

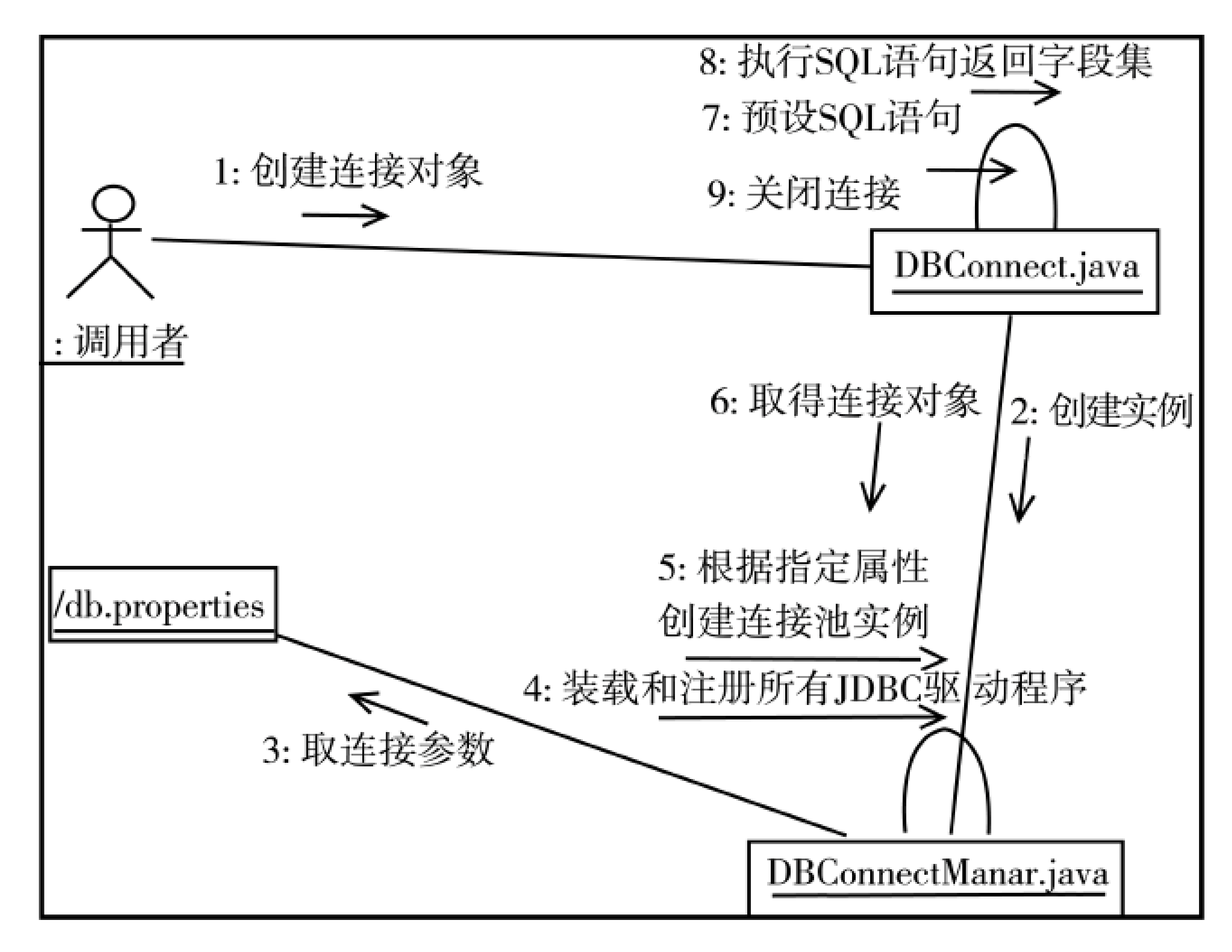


图 8-2　协作图

(5) 根据指定属性创建连接池实例。

(6) 用 getConnection( )方法取得一个连接对象。

(7) 用 PreparedStatement( )方法预设 SQL 语句。

(8) 用 executeQuery( )执行 SQL 语句。

(9) 用 close( )方法关闭连接。

8.1.1　任务描述

利用 Eclipse 实现“新闻类别网站”的数据库封装模块。



8.1.2 任务实现

本系统通过一个类 DBConnectionManager 来创建和维护一个大的连接池，在系统启动 Tomcat 时初始化该连接池。当程序需要连接数据库时，使用类 DBConnect 从该连接池中取出其中的一个空闲连接，当使用完毕时再将该连接归还给系统连接池。

在此构建了一个对数据操作的层，在这种分层体系结构中，该层是与数据库打交道的逻辑层。在数据库中对数据的增加、删除、修改和查找操作由该逻辑层来处理。初学者经常在 ASP 或 JSP 的页面代码中混杂了很多连接数据库、操作数据库的语句，这实际上是把操作数据的逻辑层和表现层耦合到了一起，使代码的编写和维护都很困难，程序结构非常不清晰。对于大型应用，更要重视这个问题。采用对系统结构分层的办法，是解决这种耦合问题的前提保障。具体分两个任务来实现。

任务 1　连接池管理类 DBConnectionManager 的实现
任务 2　连接管理类 DBConnect 的实现

任务 1：连接池管理类 DBConnectionManager 的实现

DBConnectionManager 类是对某一数据库所有连接的“缓冲池”，主要实现以下功能：①从连接池获取或创建可用连接；②使用完毕之后，把连接返还给连接池；③在系统关闭前，断开所有连接并释放连接占用的系统资源；④能够处理无效连接(原来登记为可用的连接，由于某种原因不再可用，如超时、通信问题)，并能够限制连接池中的连接总数不低于某个预定值和不超过某个预定值。具体分四个子任务来实现。

子任务 1.1　构建包结构
子任务 1.2　建立所需类的基本结构
子任务 1.3　完善所需类的基本结构
子任务 1.4　建立配置文件 db. properties

子任务 1.1：构建包结构

在 Eclipse 中新建 com. news. database 包结构，详细操作步骤参见第 7 章 7.1 节中任务实现的任务 1。

子任务 1.2：建立所需类的基本结构

在上步建立的 com. news. database 包中新建类 DBConnectionManager，保存为 DBConnectionManager. java，大致写法如程序 8-1 所示的代码。

【程序 8-1】 基本类 DBConnectionManager 的结构

Example Source Code:

```
package com.news.database;

import java.io.* ;
import java.sql.* ;
import java.util.* ;
import java.util.Date;

public class DBConnectionManager {
```






```
    static private DBConnectionManager instance;                //唯一实例
    static private int clients;

    private Vector drivers=new Vector();
    private PrintWriter log;
    private Hashtable pools=new Hashtable();

    static synchronized public DBConnectionManager getInstance() {}
    private DBConnectionManager() {}
    public void freeConnection(String name, Connection con) {}
    public Connection getConnection(String name) {}
    public Connection getConnection(String name, long time) {}
    public synchronized void release() {}
    private void createPools(Properties props) {}
    private void init() {}
    private void loadDrivers(Properties props) {}
    private void log(String msg) {}
    private void log(Throwable e, String msg) {}
}
```

子任务 1.3：完善所需类的基本结构

具体操作步骤如下：

（1）程序 8-2 用来读取数据库的配置文件 db.properties，将该配置文件中的参数保存，并调用程序 8-3 和程序 8-4 中的函数来初始化连接池。它是创建连接池的入口。首先建立读取该配置文件的连接，通过类 Properties 来加载各项配置参数，再打开日志文件写创建日志，最后分别加载数据库驱动程序，创建连接池。

【程序 8-2】 创建连接池入口

Example Source Code：

```
/**
 *读取属性完成初始化
 */
private void init() {
    InputStream is=getClass().getResourceAsStream("/db.properties");
    Properties dbProps=new Properties();
    try {
        dbProps.load(is);
    }
    catch (Exception e) {
        System.err.println("不能读取属性文件．"+"确保 db.properties 在 CLASSPATH 指定的路
径");
        return;
    }
    String logFile=dbProps.getProperty("logfile","DBConnectionManager.log");
    try {
    log=new PrintWriter(new FileWriter(logFile,true), true);
    }
    catch (IOException e) {
        System.err.println("无法打开日志文件："+logFile);
```



```
            log=new PrintWriter(System.err);
        }
        loadDrivers(dbProps);
        createPools(dbProps);
    }
```

(2)程序 8-3 用来根据读取的数据库驱动配置字符串 drivers,动态加载数据库驱动的程序。首先通过 Class.forName( )来动态加载,再使用 Drivermanager 数据库驱动管理器来注册该驱动类。

【程序 8-3】 加载数据库驱动

Example Source Code:

```
/* *
 * 装载和注册所有 JDBC 驱动程序
 * @param props 属性
 */
private void loadDrivers(Properties props) {
   String driverClasses=props.getProperty("drivers");
   StringTokenizer st=new StringTokenizer(driverClasses);
   while (st.hasMoreElements( )) {
     String driverClassName=st.nextToken( ).trim( );
     try {
       Driver driver= (Driver)
       Class.forName(driverClassName).newInstance( );
       DriverManager.registerDriver(driver);
       drivers.addElement(driver);
       log("成功注册 JDBC 驱动程序"+driverClassName);
     }
     catch (Exception e) {
       log("无法注册 JDBC 驱动程序: "+driverClassName+", 错误: "+e);
     }
   }
}
```

(3)程序 8-4 用来创建数据库连接的。首先根据配置的数据库连接字符串、数据库名、用户名、密码取得数据库的连接,如果该连接成功,则放入连接池变量 pools 中。在创建连接的过程中,需要判断参数 maxconn(连接池的创建数量),当到达该数量时,表示连接池初始化完毕。

【程序 8-4】 创建连接池实例

Example Source Code:

```
/* *
   *根据指定属性创建连接池实例
   *@param props 连接池属性
   */
private void createPools(Properties props) {
   Enumeration propNames=props.propertyNames( );
   while (propNames.hasMoreElements( )) {
   String name= (String) propNames.nextElement( );
```



```
    if (name.endsWith(".url")) {
      String poolName=name.substring(0, name.lastIndexOf("."));
      String url=props.getProperty(poolName+".url");
      if (url==null) {
        log("没有为连接池"+poolName+"指定 URL");
        continue;
      }
    String user=props.getProperty(poolName+".user");
    String password=props.getProperty(poolName+".password");
    String maxconn=props.getProperty(poolName+".maxconn", "0");
    int max;
    try {
    max=Integer.valueOf(maxconn).intValue();
    }
    catch (NumberFormatException e) {
      log("错误的最大连接数限制:"+maxconn+" .连接池:"+poolName);
      max=0;
    }
    DBConnectionPool pool=
    new DBConnectionPool(poolName, url, user, password, max);
    pools.put(poolName, pool);
    log("成功创建连接池"+poolName);
   }
 }
}
```

(4) 程序 8-5 用来从连接池对象 pools 中取得一个数据库连接。用户编写程序中，如果需要取得数据库的连接，只需调用该函数。

【程序 8-5】 获得一个可用的(空闲的)连接

Example Source Code:

```
/**
  * 获得一个可用的(空闲的)连接。如果没有可用连接,且已有连接数小于最大连接数限制,则创建
    并返回新连接
  * @param name 在属性文件中定义的连接池名字
  * @return Connection 可用连接或 null
  */
public Connection getConnection(String name) {
   DBConnectionPool pool= (DBConnectionPool) pools.get(name);
   if (pool !=null) {
     return pool.getConnection();
   }
   return null;
}
```

(5) 程序 8-6 用来释放一个连接到数据库的连接池 pools 中。在访问网站时，如果使用完某一个连接，就需要调用该函数将连接归还到连接池。

> **注意**
>
> 此项工作是必须的，因为数据库的连接有限，而且为了提高资源的利用率，不能使资源闲置，否则，久而久之系统会被空闲的、无法回收的连接耗尽内存。



【程序 8-6】 释放一个连接

Example Source Code:

```
/* *
  * 将连接对象返回给由名字指定的连接池
  * @param name 在属性文件中定义的连接池名字
  * @param con 连接对象
  * /
public void freeConnection(String name, Connection con) {
   DBConnectionPool pool= (DBConnectionPool) pools.get(name);
   if (pool !=null) {
     pool.freeConnection(con);
   }
}
```

子任务 1.4：建立配置文件 db. properties

"新闻类别系统"用连接池来配置数据库连接，数据库生成后要与网页建立动态连接，就要配置文件 db. properties，配置语句如程序 8-7 代码所示。

【程序 8-7】 配置文件 db. properties

Example Source Code:

```
drivers=com.microsoft.sqlserver.jdbc.SQLServerDriver
logfile=C:\\logN.txt

mysql.url=jdbc: sqlserver://NET-DB:1433;DatabaseName=News
mysql.maxconn=1000
mysql.user=sa
mysql.password=123456
```

Drivers 变量用来指定数据库的驱动程序为 com. Microsoft. sqlserver. SQL ServerDriver，该参数是 SQL server 2005 的特定参数。如果使用该数据库，则不能修改；如果要改用其他数据库，则需要在此做相应修改。

(1) logfile 表示系统的日志文件为 C:\\log. txt。

(2) url 表示连接数据库的连接字符串，该字符串是固定格式的，其中的参数 DatabaseName 为新建的数据库名，可以根据实际情况进行修改。

(3) maxconn 为连接池的最大数量，数值应该大于 0 才能使系统使用。

(4) user 为连接数据的用户名，password 为连接数据库的密码，这两项都可以根据需要来进行修改。

在默认的根包中新建 db. properties 文件，将程序 8-7 所示的代码输入后保存。此处要注意 db. properties 文件在包中的位置必须与 init( )中读取的位置相对应。

任务 2：连接管理类 DBConnect 的实现

DBConnect 类主要承担对数据库中记录的具体访问操作，如：执行 SQL 语句、设置字段的值等。DBConnect 类的实现具体分两个子任务来实现。

子任务 2.1 建立所需类的基本结构

子任务 2.2 完善所需类的基本结构



子任务 2.1：建立所需类的基本结构

在 8.1 节中建立的 com.news.database 包中新建类 DBConnect，保存为 DBConnect.java，大致写法如程序 8-8 代码所示。

【程序 8-8】 基本类 DBConnect 的结构

Example Source Code：

```
package com.news.database;

import java.io.PrintWriter;
import java.sql.* ;

public class DBConnect {
    private Connection conn=null;
    private Statement stmt=null;
    private ResultSet rs=null;
    private PreparedStatement prepstmt=null;
    private DBConnectionManager dcm=null;
    void init( ) {}
    public DBConnect( ) throws Exception {}
    public DBConnect(int resultSetType,int resultSetConcurrency)throws Exception {}
    public DBConnect(String sql) throws Exception {}
     public DBConnect (String sql, int resultSetType, int resultSetConcurrency) throws
     Exception {}
    public Connection getConnection( ) {}
    public void prepareStatement(String sql) throws SQLException {}
    public void prepareStatement(String sql, int resultSetType,int resultSetConcurrency)
        throws SQLException {}
    public void setString(int index, String value) throws SQLException {}
    public void setInt(int index, int value) throws SQLException {}
    public void setBoolean(int index, boolean value) throws SQLException {}
    public void setDate(int index,Date value) throws SQLException {}
    public void setLong(int index,long value) throws SQLException {}
    public void setFloat(int index,float value) throws SQLException {}
    public void setBytes(int index,byte[] value) throws SQLException {}
    public void clearParameters( ) throws SQLException {}
    public PreparedStatement getPreparedStatement( ) {}
    public Statement getStatement( ) {}
    public ResultSet executeQuery(String sql) throws SQLException {}
    public ResultSet executeQuery( ) throws SQLException {}
    public void executeUpdate(String sql) throws SQLException {}
    public void executeUpdate( ) throws SQLException {}
    public void close( ) throws Exception {}
}
```

子任务 2.2：完善所需类的基本结构

具体操作步骤如下：

(1) 在 DBConnect 类的 init( )方法中添加如程序 8-9 所示的代码，借助连接池初始化或实例化一个与数据库的连接。



【程序 8-9】 init( )方法

Example Source Code:

```
void init() {
    dcm=DBConnectionManager.getInstance();
    conn=dcm.getConnection("mysql");
}
```

(2) 在 DBConnect 类的 DBConnect( )方法中添加如程序 8-10 所示的代码,即完成 DBConnect 类的构造函数。

【程序 8-10】 基本构造函数

Example Source Code:

```
/**
*构造数据库的连接和访问类
*/
public DBConnect() throws Exception {
    init();
    stmt=conn.createStatement();
}
public DBConnect(int resultSetType,int resultSetConcurrency)throws Exception {
    init();
    stmt=conn.createStatement(resultSetType, resultSetConcurrency);
}
/**
*构造数据库的连接和访问类 预编译 SQL 语句
*@param sql        SQL 语句
*/
public DBConnect(String sql) throws Exception {
    init();
    this.prepareStatement(sql);
}
public DBConnect(String sql,int resultSetType,int resultSetConcurrency)throws Exception {
    init();
    this.prepareStatement(sql,resultSetType,resultSetConcurrency);
}
```

(3) 在 DBConnect 类的 getConnection( )方法中添加如程序 8-11 所示的代码,用于获取一个可用的连接。

【程序 8-11】 getConnection( )方法

Example Source Code:

```
/**
*返回连接
*@return Connection 连接
*/
    public Connection getConnection() {
    return conn;
}
```



(4) 在 DBConnect 类中完善用于预设 SQL 语句的 prepareStatement(String sql)方法和 prepareStatement(String sql,int resultSetType,int resultSetConcurrency)方法，具体如程序 8-12 代码所示。

【程序 8-12】　prepareStatement(String sql) 方法

Example Source Code:

```
    /**
 * PreparedStatement
 * @return sql 预设 SQL 语句
 */
public void prepareStatement(String sql) throws SQLException {
    prepstmt=conn.prepareStatement(sql);
}
public void prepareStatement(String sql,int resultSetType,int resultSetConcurrency) throws
SQLException {
    prepstmt=conn.prepareStatement(sql,resultSetType,resultSetConcurrency);
}
```

(5) 在 DBConnect 类中完善用于设置各种类型值的 setString(int index,String value)、setInt(int index,int value)、setBoolean(int index,boolean value)、setDate(int index,Date value)、setLong(int index,long value)、setFloat(int index,float value)、setBytes(int index,byte[] value)和 clearParameters( )等方法，具体的代码如程序 8-13 所示。

【程序 8-13】　设置对应值函数

Example Source Code:

```
/**
 * 设置对应值
 * @param index        参数索引
 * @param value        对应值
 */
public void setString(int index,String value) throws SQLException {
    prepstmt.setString(index,value);
}
public void setInt(int index,int value) throws SQLException {
    prepstmt.setInt(index,value);
}
public void setBoolean(int index,boolean value) throws SQLException {
    prepstmt.setBoolean(index,value);
}
public void setDate(int index,Date value) throws SQLException {
    prepstmt.setDate(index,value);
}
public void setLong(int index,long value) throws SQLException {
    prepstmt.setLong(index,value);
}
public void setFloat(int index,float value) throws SQLException {
```



```
        prepstmt.setFloat(index,value);
    }
    public void setBytes(int index,byte[] value) throws SQLException {
        prepstmt.setBytes(index,value);
    }
    public void clearParameters() throws SQLException {
        prepstmt.clearParameters();
        prepstmt=null;
    }
```

（6）在 DBConnect 类中完善用于返回预设状态的 getPreparedStatement( )方法，具体代码如程序 8-14 所示。

【程序 8-14】 getPreparedStatement( )方法

Example Source Code：

```
/**
 * 返回预设状态
 */
public PreparedStatement getPreparedStatement() {
    return prepstmt;
}
```

（7）在 DBConnect 类中完善用于返回状态的 getStatement( )方法，具体如程序 8-15 代码所示。

【程序 8-15】 getStatement( )方法

Example Source Code：

```
/**
 * 返回状态
 * @return Statement 状态
 */
public Statement getStatement() {
    return stmt;
}
```

（8）在 DBConnect 类中完善用于执行 SQL 语句的 executeQuery(String sql)方法，具体如程序 8-16 代码所示。

【程序 8-16】 executeQuery(String sql)方法

Example Source Code：

```
/**
 * 执行 SQL 语句返回字段集
 * @param sql          SQL 语句
 * @return ResultSet   字段集
 */
public ResultSet executeQuery(String sql) throws SQLException {
    if (stmt !=null) {
        return stmt.executeQuery(sql);
    } else
```



```
            return null;
    }
    public ResultSet executeQuery( ) throws SQLException {
        if (prepstmt !=null) {
            return prepstmt.executeQuery( );
        } else
            return null;
    }
```

(9) 在 DBConnect 类完善执行更新 SQL 语句的 executeUpdate(String sql)方法，具体如程序 8-17 代码所示。

【程序 8-17】 executeUpdate(String sql)方法

Example Source Code:

```
    /* *
     * 执行 SQL 语句
     * @param sql        SQL 语句
     * /
    public void executeUpdate(String sql) throws SQLException {
        if (stmt !=null)
            stmt.executeUpdate(sql);
    }
    public void executeUpdate( ) throws SQLException {
        if (prepstmt !=null)
            prepstmt.executeUpdate( );
    }
```

(10) 在 DBConnect 类中完善关闭连接的 close( )方法，具体如程序 8-18 代码所示。

【程序 8-18】 close( )方法

Example Source Code:

```
    /* *
        * 关闭连接
        * /
        public void close( ) throws Exception {
            if (stmt !=null) {
                stmt.close( );
                stmt=null;
            }
            if (prepstmt !=null) {
                prepstmt.close( );
                prepstmt=null;
            }
            if (conn !=null) {
                dcm.freeConnection("mysql", conn);
            }
        }
    }
```



8.1.3 相关知识拓展

数据库连接池技术的优势有以下几种。

1. 资源重用

由于数据库连接得到重用,避免了频繁创建、释放连接引起的大量性能开销。在减少系统消耗的基础上,另一方面也增进了系统运行环境的平稳性(减少内存碎片以及数据库临时进程/线程的数量)。

2. 更快的系统响应速度

数据库连接池在初始化过程中,往往已经创建了若干数据库连接置于池中备用。此时连接的初始化工作均已完成。对于业务请求处理而言,直接利用现有的可用连接,避免了数据库连接初始化和释放过程的时间开销,从而缩减了系统整体响应时间。

3. 新的资源分配手段

对于多应用共享同一数据库的系统而言,可在应用层通过数据库连接的配置,实现数据库连接池技术。通过设置某一应用最大可用数据库连接数的限制,可避免某一应用独占所有数据库资源。

4. 统一的连接管理,避免数据库连接泄漏

在较为完备的数据库连接池实现中,可根据预先的连接占用超时设定,强制收回被占用的连接,从而避免了常规数据库连接操作中可能出现的资源泄漏。

8.1.4 经验总结

如果是简单的数据库应用,由于对数据库的访问不是很频繁,可以简单地在需要访问数据库时,新创建一个连接,用完后就关闭它,这种方式不会带来明显的性能开销。但对于复杂的数据库应用,频繁的建立、关闭连接,会极大地降低系统的性能,因此连接的使用成了影响系统性能的瓶颈。后者的解决方法是采用连接复用。通过建立一个数据库连接池以及一套连接使用管理策略,使一个数据库连接得到高效、安全的复用,避免了数据库连接频繁建立、关闭的开销。

数据库连接池的基本原理是在内部对象池中维护一定数量的数据库连接,仅对外暴露数据库连接获取和返回方法。如:外部使用者可通过 getConnection 方法获取连接,使用完毕后再通过 releaseConnection 方法将连接返回,注意此时连接并没有关闭,而是由连接池管理器回收,并为下一次使用做好准备。

8.2　操作数据的常用命令

知识储备

SQL 是结构化查询语言(Structured Query Language)的缩写,它包括定义、操纵和控制三个部分,是一种功能齐全的数据库语言。

数据定义是指对关系模式(关系即表)的定义。数据操纵是指对关系中的具体数据进行增、删、改、查等操作。数据控制是指对数据访问权限的授予或撤销。

SQL 具有语言简洁、方便实用、功能齐全等优点。各种数据库管理系统都支持 SQL 或提供 SQL 接口。

1. SELECT 查询命令

MS SQL Server 2005 的数据库使用 T-SQL 语言,其基本的查询语句是 SELECT 语句,它是 T-SQL 语言中最基本也是最重要的语句,也是使用最频繁的查询语句。

SELECT 语句的功能十分强大,既可以实现对单表的数据查询,也可以完成复杂的多表、视图连接查询、嵌套查询等。

SELECT 语句的功能很强大,语法也很复杂,其基本语法结构如下:

```
SELECT <选择列表>
        [INTO <新表>]
        [FROM <源表>]
        [WHERE <搜索条件>]
        [GROUP BY <分组列表>]
        [HAVING <组搜索表达式>]
        [ORDER BY <排序表达式> [ASC|DESC]]
```

- <选择列表>:此语句用来设置输出到查询结果集中的列。
- [INTO <新表>]:此语句用来将查询结果集输入到一个新表中。
- [FROM <源表>]:此语句用来指定查询的源表。
- [WHERE <搜索条件>]:此语句用来指定搜索条件,只有符合搜索条件的记录才会出现在结果集中。
- [GROUP BY <分组列表>]:此语句用来指定将查询结果进行分组的条件。
- [HAVING <组搜索表达式>]:此语句在 GROUP BY 的基础之上指定筛选条件。常与 GROUP BY 搭配使用。
- [ORDER BY <排序表达式>[ASC|DESC]]:此语句用来指定结果集中记录排序的方式。

2. INSERT 插入命令

数据库中的表建好之后,可以使用 INSERT 语句向表中添加记录。添加的数据必须符合对应表中相应列的数据类型、约束等,以保证表中数据的完整性。

使用 INSERT 语句可以向已建好的表中每次添加一条记录。该语句的基本语法格式如下:

```
INSERT [INTO] <目标表名> [(列名列表)] VALUES (<值列表>)
```



- 目标表名：此语句用来指定接受数据的表名。
- 列名列表：此语句用来指定欲插入数据的各个列。这些列必须在目标表中存在，顺序可以任意；当对目标表中所有列都添加数据时，列名列表可以省略。列名列表必须用圆括号括起来。
- 值列表：此语句用来设置欲添加的数据值。各数据值与＜列名列表＞中列的顺序一一对应，且数据都必须符合相应列的数据类型、约束等，以保证表中数据的完整性。若对新记录的所有列都添加数据，并省略了列名列表，则值列表中数据的排列顺序要与目标表中列的顺序一致。值列表也要用圆括号括起来。

注意：

使用 INSERT 语句时要注意以下几点：

（1）必须为主键、不允许为空的列添加数据。

（2）添加的数据必须符合相应列的数据类型、约束，以保证表中数据的完整性。

（3）若添加的数据是 char、varchar、datetime 数据类型，则该值要用单引号括起来。

（4）对于指定有默认值的列，若要使该列获得其默认值，则在值列表的相应位置上使用 DEFAULT 关键字。

3．UPDATE 修改命令

有时需要修改表中现有的数据，在 SQL SERVER 2005 中，可以用 UPDATE 语句来实现数据修改。

UPDATE 语句可以修改表中的一条或者多条记录，其语法格式如下：

```
UPDATE <目标表名>
SET {列名=表达式|NULL|DEFAULT} [,…n]
[WHERE 搜索条件]
```

- ＜目标表名＞：此语句用来指定欲修改记录的表名。
- SET：此语句指定要修改的列和所用的数据。修改后的数据由表达式确定，但是必须要符合相应字段的数据类型、长度、精度和约束，以保证数据的完整性。另外，对具有默认值的列可以使用 DEFAULT 修改为默认值；对允许使用空值的列，也可以使用 NULL 修改为空值。
- WHERE：此语句用来筛选记录，表示 UPDATE 命令只对符合搜索条件的记录进行修改，当省略 WHERE 子句时，将对表中的所有记录都修改。

4．DELETE 修改命令

对于表中已经确定无用的记录，可以用 DELETE 语句删除。

DELETE 语句可以从表中删除一条或者多条记录，语法格式如下：

```
DELETE [FROM] <目标表名>
[WHERE <搜索条件>]
```

- FROM：此语句是可选参数。
- ＜目标表名＞：此语句指定要从中删除记录的表名。
- WHERE：此语句用来指定筛选条件，将满足＜搜索条件＞的记录进行删除。该子句为可选：若不选，则表示删除表中的所有记录；若选，则删除满足条件的记录。



8.2.1　任务描述

使用 T-SQL 编写数据操作命令。

8.2.2　任务实现

在每个系统中，都需要对字符串、页面参数、网页 HTML 代码、Cookie 参数进行处理来实现系统的交互，可以说对系统的实现都是在对字符串进行格式化处理的基础之上。因此，根据网页整体设计的功能抽象，需要对各种类型的字符串进行特殊的输入输出处理，下面列出了如下功能的组件设计。具体操作步骤如下：

(1) 编写查询数据的操作命令

"Com.news.menu"包中的 MenuManage 类的 getMenu( )方法，其功能是获取所有的新闻分类，并在调用时对"新闻分类"按照 ID 进行排序。如程序 8-19 所示。

【程序 8-19】 getMenu( )方法

Example Source Code:

```
public static Vector getMenu( ) throws MenuNotFoundException {
    try {
        DBConnect dbc=new DBConnect("select * from Menu order by id");
        ResultSet rs=dbc.executeQuery( );
        Vector menuVector=new Vector( );
        while (rs.next( )) {
            Menu tempMenu=new Menu( );

            tempMenu.setMenuID(rs.getInt(1));
            tempMenu.setMenuName(rs.getString(2));
            tempMenu.setMenuEName(rs.getString(3));
            tempMenu.setMenuMe(rs.getString(4));
            tempMenu.setMenuURL(rs.getString(5));
            tempMenu.setAddTime(rs.getString(6));
            tempMenu.setOrderID(rs.getInt(7));
            tempMenu.setParentID(rs.getInt(8));
            tempMenu.setHide(rs.getInt(9));

            menuVector.add(tempMenu);
        }
        dbc.close( );
        return menuVector;
    } catch (Exception e) {
        e.printStackTrace( );
        throw new MenuNotFoundException( );
    }
}
```

(2) 编写添加数据的操作命令

"Com.news.menu"包中的 MenuManage 类的 saveMenu(HttpServletRequest request)



方法，其功能是添加一个新的新闻分类，并对必要的信息进行判断，如程序 8-20 代码所示。

【程序 8-20】 saveMenu(HttpServletRequest request)方法

Example Source Code:

```
public static void saveMenu(HttpServletRequest request) throws Exception {

    String MenuName=ParamUtil.getString(request,"MenuName");
    String MenuEName=ParamUtil.getString(request,"MenuEName");

    if (MenuName==null || MenuEName==null)
        throw new Exception("请输入完整新闻分类信息。");
    DBConnect dbc=new DBConnect("insert into Menu(MenuName,MenuEName,MenuMe,
        MenuURL,AddTime,OrderID,ParentID) values(?,?,?,?,?,?,?)");

    dbc.setString(1,new String(ParamUtil.getString(request, "MenuName","").getBytes
        ("ISO-8859-1"), "GBK"));
    dbc.setString(2,ParamUtil.getString(request, "MenuEName",""));
    dbc.setString(3,new String(ParamUtil.getString(request, "MenuMe","").getBytes
        ("ISO-8859-1"), "GBK"));
    dbc.setString(4,new String(ParamUtil.getString(request, "MenuURL","").getBytes
        ("ISO-8859-1"),"GBK"));
    dbc.setString(5,Format.getDateTime());
    dbc.setInt(6,ParamUtil.getInt(request, "OrderID",1));
    dbc.setInt(7,ParamUtil.getInt(request, "ParentID",1));

    dbc.executeUpdate();
    dbc.close();
}
```

(3) 修改数据

“Com. news. menu”包中的 MenuManage 类的 saveEdit (HttpServletRequest request) 方法，其功能是修改一个新闻类别，调用此函数前要先选定想要修改的新闻类别，如程序 8-21代码所示。

【程序 8-21】 saveEdit(HttpServletRequest request)方法

Example Source Code:

```
public static void saveEdit(HttpServletRequest request) throws Exception {
    int MenuID;
    try {
        MenuID=ParamUtil.getInt(request, "MenuID");
    } catch (Exception e) {
        throw new Exception("请选择论坛的 ID.");
    }

    DBConnect dbc=new DBConnect("update Menu set MenuName=?,MenuEName=?,MenuMe=?,
        MenuURL=?,OrderID=?,ParentID=?where id="+MenuID);

    dbc.setString(1,new String(ParamUtil.getString(request,"MenuName","").getBytes
        ("ISO-8859-1"), "GBK"));
```



```
        dbc.setString(2,ParamUtil.getString(request, "MenuEName",""));
        dbc.setString(3,new String(ParamUtil.getString(request,"MenuMe","").getBytes
            ("ISO-8859-1"), "GBK"));
        dbc.setString(4,new String(ParamUtil.getString(request,"MenuURL","").getBytes
            ("ISO-8859-1"),"GBK"));
        dbc.setInt(5,ParamUtil.getInt(request, "OrderID",1));
        dbc.setInt(6,ParamUtil.getInt(request, "ParentID",1));

        dbc.executeUpdate();

        dbc.close();
    }
```

（4）删除数据

"Com. news. menu"包中的 MenuManage 类的 delMenu(int MenuID)方法，其功能是删除一个新闻分类，同时查找该分类下的子分类一并删除，调用此函数前先选择要删除的新闻分类 ID，如程序 8-22 代码所示。

> **注意**
>
> 在此要注意对相关类别下新闻的删除，因为删除了新闻分类，那么相关新闻的对应分类就不能确定了，因此必须考虑将分类下的所有新闻删除。这里不给出相应的代码，读者可以自行思考、设计。

【程序 8-22】delMenu(int MenuID)方法

Example Source Code:

```
public static void delMenu(int MenuID) {

    try {
        int i;
        String sql="";

        Vector menuVector=new Vector();
        menuVector=getMenus(MenuID);

        for (i=0; i<menuVector.size(); i++) {
            Menu tempMenu=(Menu) menuVector.get(i);
            delMenu(tempMenu.getMenuID());
        }

        Vector articleVector=new Vector();
        articleVector=ArticleManager.getArticles(MenuID);

        for (i=0; i<articleVector.size(); i++) {
            Article tempArticles=(Article) articleVector.get(i);
            ArticleManager.delArticle(tempArticles.getId());
        }
        DBConnect dbc=new DBConnect();
        sql="delete from Menu where id="+MenuID;
```



```
            dbc.executeUpdate(sql);

            dbc.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

8.2.3 相关知识拓展

SQL 是由命令、子句和运算符等元素所构成的，这些元素结合起来组成用于创建、更新和操作数据库的语句。

1. SQL 命令

SQL 命令常分为三类，数据定义语言 DDL(Data Definition Language)、数据操纵语言 DML(Data Manipulation Language)、数据控制语言 DCL(Data Control Language)。

(1) 数据定义语言

DDL 定义数据库的结构，包括数据库本身、数据表、目录、视图等数据库元素。常用的 DDL 语句及其相关说明如表 8-1 所示。

表 8-1 常用 DDL 语句及其相关说明

常用 DDL 语句	说　明
CREATE table	创建表
CREATE index	创建索引
CREATE view	创建视图
ALTER table	修改表定义：包括增加表列、重定义表列、删除表列等
DROP table	删除表定义
DROP index	删除索引

(2) 数据操纵语言

DML 完成在数据库中确定、修改、添加、删除某一数据值的任务。常用的 DML 语句及其相关说明如表 8-2 所示。

表 8-2 常用 DML 语句及其相关说明

常用 DML 语句	说　明
SELECT	在数据库中查找满足指定条件的记录
INSERT	在表中增加数据行
DELETE	从表中删除数据行
UPDATE	更改表中数据





(3) 数据控制语言

DCL 用来授予或回收访问数据库的某种特权，并控制数据库操纵事务发生的时间及效果，对数据库实行监视等。常用的 DCL 语句及其相关说明如表 8-3 所示。

表 8-3　常用 DCL 语句及其相关说明

DCL 语句	说　　明
GRANT	将权限或角色授予用户或其他角色
REVOKE	回收用户权限
ROLL	回滚(把信息恢复到用户使用 update、insert、delete 前最后提交的状态)
COMMIT	提交(在完成数据库的插入，删除和修改操作时，只有当事务提交到数据库才算完成)

2. 子句

一个 SQL 语句是一个由子句组成的命令，子句指定了要执行的操作、数据源以及任何完成操作所需的结构。每个从句必须起始于一个关键字。最常用的从句及其相关说明如表 8-4 所示。

表 8-4　常用 DCL 语句及其相关说明

子　　句	说　　明
FROM	指定要对其进行操作的数据源
WHERE	对操作设定一个或多个条件
ORDER BY	对查询结果进行排序
GROUP BY	对查询结果进行分组
HAVING	指定分组的条件

3. 运算符

SQL 运算符可分为下面几种类别。

- 比较运算符(大小比较)：＞、＞＝、＝、＜、＜＝、＜＞、!＞、!＜
- 范围运算符(表达式值是否在指定的范围)：BETWEEN… AND…、NOT BETWEEN…AND…
- 列表运算符(判断表达式是否为列表中的指定项)：IN (项 1，项 2…)、NOT IN (项 1，项 2…)
- 模式匹配符(判断值是否与指定的字符通配格式相符)：LIKE、NOT LIKE
- 空值判断符(判断表达式是否为空)：IS NULL、IS NOT NULL
- 逻辑运算符(用于多条件的逻辑连接)：NOT、AND、OR

SQL 还有一些用来计算的函数，称为聚合函数。例如，用 AVG 函数计算平均值，用 COUNT 函数返回记录数，用 SUM 函数计算总和，用 MAX 函数计算最大值等。



8.2.4 经验总结

SQL 是实现数据库查询性能提升的最现实、最有效的手段，有时甚至是唯一的手段。下面列出通过优化 SQL 语句提高查询性能的方法。

1. SQL 书写规范

(1) SQL 语句的所有表名、字段名全部小写，系统保留字、内置函数名、SQL 保留字大写。

(2) 连接符 OR、IN、AND，以及=、<=、>=等，前后加上一个空格。

(3) 对较为复杂的 SQL 语句加上注释，说明算法、功能。

- 对不易理解的分支条件表达式加注释。
- 重要的计算应说明其功能。
- 过长的函数实现，应将其语句按实现的功能分段加以概括性说明。
- 每条 SQL 语句均应有注释说明（表名、字段名）。
- 常量及变量注释时，应注释被保存值的含义（必须）、合法的取值范围（可选）。
- 可采用单行/多行注释（--或/* */方式）。

(4) SQL 语句的缩进风格

- 一行有多列，超过 80 个字符时，基于列对齐原则，采用下行缩进。
- WHERE 子句书写时，每个条件占一行，语句另起一行时，以保留字或者连接符开始，连接符右对齐。

(5) 多表连接时，使用表的别名来引用列。

(6) 供别的文件或函数调用的函数，绝不应使用全局变量交换数据。

2. 书写优化性能建议

(1) 避免嵌套连接。例如，A=B and B=C and C=D。

(2) WHERE 条件中尽量减少使用常量比较，改用变量。

(3) 系统可能选择基于规则的优化器，所以将结果集返回数据量小的表作为驱动表（FROM 后边最后一个表）。

(4) 大量的排序操作影响系统性能，所以尽量减少 ORDER BY 和 GROUP BY 排序操作。如必须使用排序操作，应遵循如下规则：

- 排序尽量建立在有索引的列上。
- 如结果集不需唯一，使用 UNION ALL 代替 UNION。

(5) 尽量避免相同语句由于书写格式的不同，而导致多次语法分析。

(6) 尽量使用共享的 SQL 语句。

(7) 查询中的 WHERE 过滤原则，应使过滤记录数最多的条件放在最前面。

(8) 任何对列的操作都将导致表扫描，它包括数据库函数、计算表达式等，查询时要尽可能将操作移至等号右边。

3. 其他经验性规则

(1) 尽量少用嵌套查询。



(2) 用多表连接代替 EXISTS 子句。

(3) 少用 DISTINCT 子句。

(4) 使用 ROWID 提高检索速度。对 SELECT 得到的单行记录，需进行 DELETE、UPDATE 操作时，使用 ROWID 将会使效率大大提高。

(5) 使用优化线索机制进行访问路径控制。

8.3　管理员的登录

8.3.1　任务描述

利用 Eclipse 实现“新闻类别网站”的管理员登录模块。

8.3.2　任务实现

登录页面具有自动识别网站访问者身份类别的功能，不同访问者登录时将根据其不同的身份进入不同的功能页面。系统共设三种访问者级别，在登录身份验证通过后，系统利用 Session 变量记录登录名、身份类别，并据此分别进入相应的模块，伴随这个访问者对网站进行操作的整个生命周期。“新闻类别网站”登录模块的具体实现步骤如下：

(1) 在“com. news. util”包中新建基本类 Login，保存为 Login. java，大致写法如程序 8-23所示。

【程序 8-23】 基本类 Login 的结构

Example Source Code：

```
package com.news.util;          // 指定类所在的包

import java.sql.* ;
import javax.servlet.http.HttpServletRequest;

import com.news.util. * ;
import com.news.database. * ;

public class Login {
    Function Fun=new Function( );
    MD5 md5=new MD5( );
    public static int AdminID;
    public static int AdminType;
    public static int LoginNum;

    public Login( ) {}
    public boolean loginCheck(HttpServletRequest request) {}
    public static void upLogin(String ip,int id,int LoginNum) throws Exception {}
}
```

(2) 在新建的基本类 Login 中，完善用于验证登录的 loginCheck(HttpServletRequest



request)方法,大致写法如程序 8-24 所示。

【程序 8-24】 loginCheck(HttpServletRequest request)方法

Example Source Code:

```
/*****************************************************************************
 * 函数名: Check
 * 作用: 验证登录
 * 参数: s1,s2,s3: 字符串型,登录用户名,密码,IP
 * 返回值: 布尔型。登录成功返回 True,否则返回 False
 * @ throws Exception
*****************************************************************************/
public boolean loginCheck(HttpServletRequest request) {
    try {
        boolean OK=false;
        String User=ParamUtil.getString(request,"User","");
        String Pwd=ParamUtil.getString(request,"Pwd","");
        DBConnect dbc=new DBConnect("SELECT * FROM Admin WHERE AdminName='"+User
                  +"' AND AdminPwd='"
                  +md5.getMD5ofStr(md5.getMD5ofStr(Pwd))+"'");
        ResultSet rs=dbc.executeQuery();
        if (rs.next()) {
            AdminID=rs.getInt("ID");
            LoginNum=rs.getInt("LoginNum")+1;
            upLogin(request.getRemoteHost(),AdminID,LoginNum);
            OK=true;
        }
        dbc.close();
        return OK;
    } catch (Exception e) {
        return false;
    }
}
```

(3) 在新建的基本类代码的 Login 中,完善更新登录信息的 upLogin(String ip,int id,int LoginNum)方法,大致写法如程序 8-25 所示。

【程序 8-25】 upLogin(String ip,int id,int LoginNum)方法

Example Source Code:

```
/*****************************************************************************
 * 函数名: upLogin
 * 作用: 更新登录信息
 * 参数: s1,s2,s3: 字符串型,登录用户名,密码,IP
 * 返回值: 布尔型。登录成功返回 True,否则返回 False
 * @throws Exception
*****************************************************************************/
public static void upLogin(String ip,int id,int LoginNum) throws Exception {
    DBConnect dbc=new DBConnect();
    String sql="UPDATE Admin SETLastLoginTime='"+Format.getDateTime()+"' WHERE id="+id;
    dbc.executeUpdate(sql);
```



```
        sql="UPDATE Admin SET LastLoginIP='"+ip+"' WHERE id="+id;
        dbc.executeUpdate(sql);
        sql="UPDATE Admin SET LoginNum='"+LoginNum+"' WHERE id="+id;
        dbc.executeUpdate(sql);
        dbc.close();
    }
```

(4) 如图 8-3 所示的登录界面，是在表现层的公共页面的基础上制作的，所以，在 Eclipse 中打开一个 Temp-jsp.jsp 文件，另存为 Login.jsp，作为管理员登录的 JSP 模板。



图 8-3　登录界面

(5) 在新建的登录页面 Login.jsp 中，完善登录页面的表单和相关的验证方法，大致写法如程序 8-26 所示。

【程序 8-26】　upLogin 方法

Example Source Code：

```
<%@page contentType="text/html;charset=GB2312" pageEncoding="GBK"%>
<%@page import="com.news.util.*"%>
<%@page import="com.news.user.*"%>
<%@page import="com.news.config.*"%>
<%
    try{
    response.setHeader("Pragma","No-cache");
    response.setHeader("Cache-Control","no-cache");
    response.setDateHeader("Expires", 0);
    request.setCharacterEncoding("GBK");

    Config con=new Config();
    out.print(con.PrintHead("登录页面"));
```



```
    String Action=request.getParameter("Action");

    if(Action!=null && Action.equals("Login"))
    {
        //得到页面地址来源
        String Page1=(String)request.getHeader("Referer");
        //得到当前页面地址
        String Page2=request.getRequestURL().toString();
        String methon=request.getMethod();
        if (methon.equals("POST") && Page2.equals(Page1))
        {
            try{
            Function Fun=new Function();
            Login login=new Login();

            String sCode=Fun.CheckReplace(request.getParameter("VerifyCode"));

            if(!sCode.equals(Session.getAttribute("VerifyCode")))
        {
            Session.setAttribute("error","<li>验证码错误");
            response.sendRedirect("Login.jsp");
        }
        if (login.loginCheck(request))
        {
            Session.setAttribute("Login","Yes");
            Session.setAttribute("AdminID",Integer.toString(login.AdminID));
            out.println("<SCRIPT LANGUAGE='JavaScript'>alert('登录成功!');location.href='
            Admin_Main.jsp';</SCRIPT>");
        }
        else
        {
            Session.setAttribute("error","<li>用户名或密码错误");
            response.sendRedirect("Login.jsp");
        }
    }catch(Exception e){
        response.sendRedirect("Login.jsp");
    }
        }
        else
        {
    out.print("出错"+Page1+";"+Page2);
    return;
        }
    }
%>

<div id="Main">
<div align="center"><%=((String)Session.getAttribute("error")==null)? "":  (String)
Session.getAttribute("error")%></div>
<form name="form" action="" method="POST" onSubmit="return LoginCheck()">
<table border="0" align="center" cellpadding="2" cellspacing="2">
```



```
        <tr>
            <th colspan="2">用户登录</th>
        </tr>
        <tr>
            <td>用户名：</td>
            <td><input name="User" type="text" id="User" size="16" /></td>
        </tr>
        <tr>
            <td>密 码：</td>
            <td><input name="Pwd" type="password" id="Pwd" size="16" /></td>
        </tr>
        <tr>
            <td>验证码：</td>
            <td>
            <input name="VerifyCode" type="text" id="VerifyCode" size="6" maxlength="6" />
            <img src="VerifyCode.jsp" alt=""> (区分大小写)
            </td>
        </tr>
        <tr>
            <td colspan="2" align="center" height="30">
            <input type="submit" name="submit" value="登 录" class="button">
            <input type="reset" name="submit2" value="清 除" class="button">
            </td>
        </tr>
    </table>
    </form>
    </div>
    <%
        out.print(con.PrintFooter());
    }
    catch(Exception e){
        e.printStackTrace();
        String errMsg=e.getMessage();
    }%>
```

8.3.3　相关知识拓展

Session tracking(会话追踪)是一类用来在客户端与服务器之间保持状态的解决方案，简单地说，当一个客户在多个页面间切换时，服务器会保存该用户的信息。会话追踪的实现方式有下列 4 种方式：

(1) 使用持续 Cookies(Persistent Cookies)。

(2) 重写包含额外参数的 URL(URL Rewriting)。

(3) 建立含有数据的隐藏表单字段(Hidden Form Field)。

(4) 使用内建 Session 对象。

前三种会话追踪方式是传统的做法，每种做法都有缺点。最后一种方法是目前最常用的，也是最有效的解决方案，因此在本节中将把讨论重心放在第 4 种会话追踪方式上。然而



为彻底了解会话追踪的机制，还是先将传统的会话追踪方式做一下介绍。

1. 使用 Cookie

Cookie 是一个小小的文本文件，它将会话信息记录在这个文本文件内，每个页面都在 Cookie 中提取以前的会话信息。例如：

```
String SessionID=makeUniqueString();
HashMap SessionInfo=new HashMap();
HashMap globalTable=findTableStoringSessions();
globalTable.put(SessionID, SessionInfo);
Cookie SessionCookie =new Cookie("JSESSIONID", SessionID);
SessionCookie.setPath("/");
response.addCookie(SessionCookie);
```

上面这段代码先将会话信息记录在 HashMap 中，保存在服务器端，并用 SessionID 标识，然后把 SessionID 保存在名为“JSESSIONID”的 Cookie 中。

访问者的请求到达服务器后，先从 Cookie 中取出 SessionID，然后从 HashMap 中取出会话信息。这样就实现了会话追踪。

虽然 Cookie 强大且持续性高，但是由于有些访问者因为担心 Cookie 对个人隐私的威胁，会关闭 Cookie，一旦如此，便无法利用 Cookie 来达到会话追踪的功能。

2. URL 重写

URL 重写是利用 GET 方法，在 URL 的尾部添加一些额外的参数来达到会话追踪(Session tracking)的目的，服务器将这个标识符与它所存储的有关会话的数据关联起来。下面是一个 URL 的写法示例：

```
http://host/path/file.html;jSessionid=1234
```

使用 URL 重写的优点是在 Cookie 被禁用或者根本不支持的情况下，网站依旧能够实现会话跟踪。但这种方式也有很多缺点，比如必须对所有指向网站的 URL 进行编码；所有页面必须动态生成；不能使用预先记录下来的 URL 进行访问，或者从其他网站链接进行访问。

3. 隐藏表单字段

隐藏表单字段的方法，是利用 HTML 的表单字段的 Hidden 属性，把客户端的信息在访问者不察觉的情形下，偷偷地随着客户端请求一起传送给服务器进行处理，这样一来，就可以进行会话跟踪的任务了。可用下面的方法利用隐藏表单字段来实现会话追踪。

```
<input type="Hidden" name="userID" value="15">
```

这样就可以将重要的客户端信息，如 ID 之类的独一无二的数据，以隐藏字段的方式传送给服务器。隐藏字段的优点在于 Session 数据传送到服务器端时，并不像 GET 方法会将 Session 数据暴露在 URL 之中。不过这种做法也有缺点，只要 Session 数据储存在隐藏字段中，就会有暴露数据的危机，因为只要直接观看 HTML 源文件，Session 数据将会暴露无疑。这将造成安全上的漏洞，特别当访问者数据是依赖于 ID、密码来取得的时候，将会有被盗用的危险。



4. 使用内建 Session 对象

传统的会话追踪方式使用起来比较麻烦，但 JSP 使用内建的 Session 对象可以非常方便地实现会话追踪，JSP 的会话机制基于 Cookie 或 URL 重写技术，融合了这两种技术的优点，当客户端允许使用 Cookie 时，内建 Session 对象使用 Cookie 进行会话追踪，如果客户端禁用 Cookie，则选择使用 URL 重写。

(1) 获取 Session 对象

例如把购物车作为属性存储在 Session 中，在其他 JSP 页面中可以通过 getAttribute 方法读取 Session 中的对象获得购物车。

```
//在 JSP 页面中可以直接使用 Session
ShoppingCart cart= (ShoppingCart)Session.getAttribute("cart");
```

内建的 Session 对象是 javax. servlet. http. HttpSession 类的实例，如果在 JavaBean 或者 Servlet 中使用 Session，就需要先从当前的 request 对象中取得，例如：

```
//得到用户 Session 和购物篮
HttpSession Session=request.getSession( );
ShoppingCart cart= (ShoppingCart)Session.getAttribute("cart");
```

(2) 读写 Session 中的数据

向 Session 中存入对象时使用 setAttribute 方法。注意，从 Session 返回的值要转换成合适的类型，要注意检查结果是否为 null。例如下面一段代码：

```
HttpSession Session=request.getSession( );
SomeClass value= (SomeClass)Session.getAttribute("someID");
if (value==null) {
value=new SomeClass(...);
Session.setAttribute("someID", value);
}
doSomethingWith(value);
```

(3) 废弃 Session 数据

调用 removeAttribute 废弃 Session 中的值，即移除与名称关联的值。

调用 invalidate 废弃整个 Session，即废弃当前的 Session。

如果访问者注销并离开站点，注意要废弃与他相关联的所有 Session。

(4) Session 的生命周期

由于没有办法知道 HTTP 客户端是否不再需要 Session，因此每个 Session 都关联一个时间期限，使它的资源可以被回收。

(5) 服务器使用 URL 重写

Session 默认使用 Cookie 技术进行会话追踪，如果客户端不接受 Cookie 的时候，服务器可以利用 URL 重写的方式将 SessionID 作为参数附在 URL 后面，此时须利用 response 对象内的 encodeURL 或 encodeRedirectURL 方法，这两个方法首先判断 Cookies 是否被浏览器支持，如果支持，则参数 URL 被原样返回，SessionID 将通过 Cookies 来维持。否则返回带有 SessionID 的 URL。



8.3.4 经验总结

登录验证在 Web 应用程序设计中占有相当重要的地位，是一个很重要的模块。一般来说，访问者需要提供登录名和密码，发出登录请求，经过密码验证模块的验证后，将把该访问者的一些特定信息保存到 Session 中，并进行日志记录和权限的设置，当访问者浏览其他设有特定权限的页面时，他的相关信息就需要在不同的页面间实现共享。具体处理如图 8-4 所示。

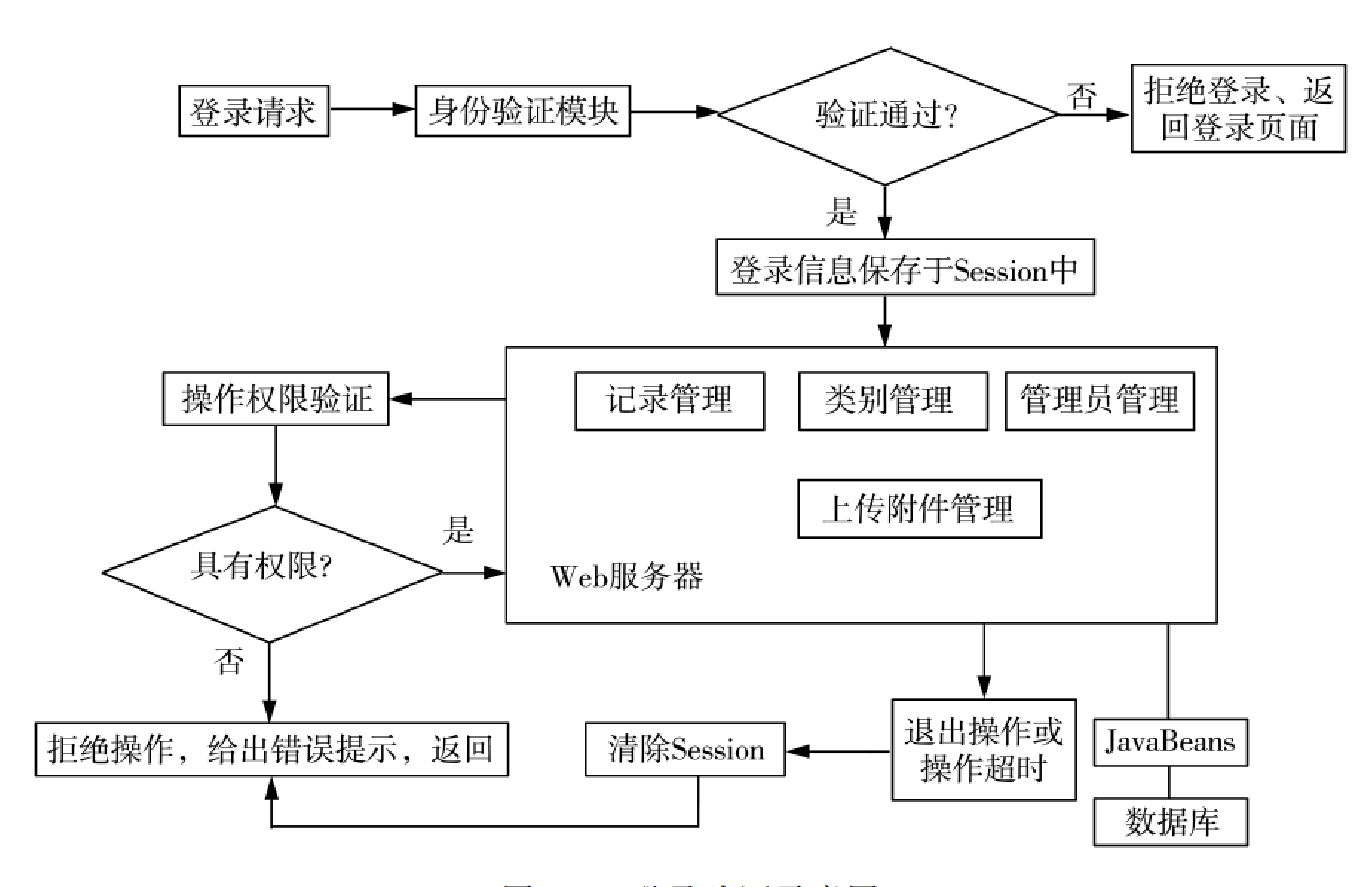


图 8-4　登录验证示意图

8.4 新闻的添加、修改、删除

8.4.1 任务描述

利用 Eclipse 实现“新闻类别网站”新闻管理模块的数据添加、修改、删除等操作。

8.4.2 任务实现

一般的新闻组件维护需要的操作有增加新闻、修改新闻、删除新闻、查询所有的新闻以及查询某一个新闻的属性数据。本节主要完成新闻的增、删、改功能。“新闻类别网站”的这些功能具体分三个任务来实现。

任务 1　实现添加操作
任务 2　实现修改操作
任务 3　实现删除操作



任务 1：实现添加操作

具体操作步骤如下：

（1）打开"com. news. article"包中的 ArticleManage 类，在保存新闻信息的 saveArticle (HttpServletRequest request)方法中添加如程序 8-27 所示的代码。

此方法的功能是添加一条新闻，调用函数时会对"新闻名称"、"新闻分类"、"新闻说明"、"新闻开放状态"等数据进行非空判断和唯一性判断，如果为空时抛出异常，并给出操作者相应的出错提示。新闻添加成功则会在数据库的新闻表里增加一条记录。

【程序 8-27】 saveArticle(HttpServletRequest request) 方法

Example Source Code：

```
public static void saveArticle(HttpServletRequest request) throws Exception {

    DBConnect dbc= new DBConnect("insert into Article(ArticleTitle, ArticleInfo,
      ArticleContent, ArticleKey, ArticleAuthor, ArticleFrom, ArticleTime, ArticlePicture, MenuID,
      IsHead, IsImg, IsHot, SpecialID,AdminName)values(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");

    dbc.setString(1,new String(ParamUtil.getString(request, "ArticleTitle","").
      getBytes("ISO-8859-1"),"GBK"));
    dbc.setString(2,new String(ParamUtil.getString(request, "ArticleInfo", "").
      getBytes("ISO-8859-1"),"GBK"));
    dbc.setString(3,new String(ParamUtil.getString(request, "ArticleContent","").
      getBytes("ISO-8859-1"), "GBK"));
    dbc.setString(4,new String(ParamUtil.getString(request, "ArticleKey", "").
      getBytes("ISO-8859-1"),"GBK"));
    dbc.setString(5, new String(ParamUtil.getString(request, "ArticleAuthor","").
      getBytes("ISO-8859-1"),"GBK"));
    dbc.setString(6, new String(ParamUtil.getString(request, "ArticleFrom", "").
      getBytes("ISO-8859-1"),"GBK"));
    dbc.setString(7, Format.getDateTime());
    dbc.setString(8, ParamUtil.getString(request, "ArticlePicture", ""));
    dbc.setInt(9, ParamUtil.getInt(request, "MenuID", 0));
    dbc.setInt(10, ParamUtil.getInt(request, "IsHead", 0));
    dbc.setInt(11, ParamUtil.getInt(request, "IsImg", 0));
    dbc.setInt(12, ParamUtil.getInt(request, "IsHot", 0));
    dbc.setInt(13, ParamUtil.getInt(request, "SpecialID", 0));
    dbc.setString(14, new String(ParamUtil.getString(request, "AdminName","").
      getBytes("ISO-8859-1"), "GBK"));

    dbc.executeUpdate();
    dbc.close();
}
```

（2）打开第 7 章中建立的 Admin_Articles. jsp 文件，完善对应的添加界面的设计部分，所需代码如程序 8-28 所示。此处代码用于产生添加界面，即用于提交数据的表单页面，而并不是将数据进行处理的业务程序。

【程序 8-28】 新闻管理模块表现层的实现



Example Source Code:

```
    ⋮
    } else if (Action.equals("add")) {
%>
<form name="AddNews" method="POST" action="?Action=save">
    <table>
    <tr>
        <td align="right">所属大类:</td>
        <td colspan="2">
        <select name="ClassID" size="1" id="ClassID"
    onChange="ChangeLocation(document.AddNews.ClassID.options[document.AddNews.
      ClassID.selectedIndex].value)">
    <%
        int i;
        Vector ClassVector=MenuManage.getMenus(0);
        if (ClassVector.size()>0) {
            out.println("<option value=\"\">请选择大类</option>");
            for (i=0; i<ClassVector.size(); i++) {
                Menu tempClass=(Menu) ClassVector.get(i);
                    out.print("<option value=\""+tempClass.getMenuID()+"\">"
                      +tempClass.getMenuName()+"</option>");
            }
        } else {
            out.println("<option value=\"\" selected>暂无大类</option>");
        }
    %>
        </select>
        <span class="info1">*请选择新闻的类别</span>
        <select name="SpecialID" size="1" id="SpecialID"
         onChange="ChangeLocation(document.AddNews.BigClassID.options[document.AddNews.
         BigClassID.selectedIndex].value)">
            <option selected value="">不属于任何专题</option>
        </select>
        </td>
    </tr>
    <tr>
        <td align="right">标题:</td>
        <td colspan="2">
        <input name="NewsTitle" type="text" id="NewsTitle" size="40" maxlength="60">
        </td>
    </tr>
    <tr>
        <td align="right">发布人:</td>
        <td width="120"><input name="NewsAuthor" type="text"
            id="NewsAuthor" size="10" maxlength="20"></td>
        <td>出处:<input name="NewsFrom" type="text" id="NewsFrom"
            size="10" maxlength="60" /></td>
    </tr>
    <tr>
        <td align="right">关键字:</td>
        <td colspan="2"><input name="NewsKey" type="text" id="NewsKey"
            size="40" maxlength="30"></td>
```






```
</tr>
<tr>
    <td align="right">图片新闻：</td>
    <td><label><input type="radio" name="IsImg" value="1"
        onClick="ShowPicture(true)">是</label><label><input
        name="IsImg" type="radio" value="0" checked
        onClick="ShowPicture(false)">否</label></td>
    <td>
    <div id="p" style="display: none">  <input
        name="NewsPicture" type="text" id="NewsPicture" size=25
        maxlength="80" style="display: " value=""><input
        name="UpImg" type="button" id="UpImg" onClick="getPicture()"
        value="新闻图片>>"> <span class="info1">(130*80)</span></div>
    </td>
</tr>
<tr>
    <td align="right">头条新闻：</td>
    <td><label><input type="radio" name="IsHead" value="1"
        onClick="ShowHeadPicture(true)">是</label><label><input
        name="IsHead" type="radio" value="0" checked
        onClick="ShowHeadPicture(false)">否</label></td>
    <td>
    <div id="p1" style="display: none">  <input
        name="HeadPicture" type="text" id="HeadPicture" size=25
        maxlength="80" style="display: " value=""><input
        name="UpImg" type="button" id="UpImg" onClick="getHeadPicture()"
        value="头条图片>>"> <span class="info1">(250*140)</span></div>
    </td>
</tr>
<tr>
    <td align="right">滚动新闻：</td>
    <td colspan="2"><label><input name="IsHot" type="radio"
        value="1">是</label><label><input name="IsHot" type="radio"
        value="0" checked>否</label></td>
</tr>
<tr>
    <td align="right">新闻摘要：</td>
    <td colspan="2"><textarea name="NewsInfo" cols="40" rows="3"
        id="NewsInfo"></textarea><span class="info1">*120字以内</span></td>
</tr>
<tr>
    <td colspan="3">
    <FCK:editor id="NewsContent"
        basePath="/FCKeditor/" width="600" height="400"
        skinPath="/FCKeditor/editor/skins/silver/" defaultLanguage="zh-cn"
        tabSpaces="8"
 imageBrowserURL="/FCKeditor/editor/filemanager/browser/default/browser.html?Type=
Image&Connector=connectors/jsp/connector"
linkBrowserURL="/FCKeditor/editor/filemanager/browser/default/browser.html?Connector
=connectors/jsp/connector"
 flashBrowserURL="/FCKeditor/editor/filemanager/browser/default/browser.html?Type=
Flash&Connector=connectors/jsp/connector"
imageUploadURL="/FCKeditor/editor/filemanager/upload/simpleuploader?Type=Image"
```



```
    linkUploadURL="/FCKeditor/editor/filemanager/upload/simpleuploader?Type=File"
    flashUploadURL="/FCKeditor/editor/filemanager/upload/simpleuploader?Type=Flash">
        </FCK:editor>
        </td>
    </tr>
    <tr>
        <td colspan=3 align="center"><input name="B1" type="button"
            value="返回>>" onClick="location.href='ListNews.jsp'"><input
            name="AdminName" type="hidden" id="AdminName"
            value="<%=new User(Integer.parseInt(Session.getAttribute(
                                "AdminID").toString())).getName()%>">
         <input name="B2" type="button" id="B2" value="增加>>"
            onClick="CheckNews();"></td>
    </tr>
</table>
    </form>
<%
```

(3) 打开第 7 章中建立的 Admin_Articles.jsp 文件，完善对应的处理添加操作的部分，所需代码如程序 8-29 所示。此处代码用于处理数据添加操作，即用于进行数据处理的业务层，将调用程序 8-27 所示的 ArticleManage 类的 saveArticle(HttpServletRequest request)方法，把从表单中接收的数据写入数据库中。

【程序 8-29】 新闻管理模块表现层的实现

Example Source Code:

```
⋮
} else if (Action.equals("save")) {
    ArticleManager.saveArticle(request);
    out.println("save 操作成功!");
    out.println("<SCRIPT LANGUAGE='JavaScript'>alert('del 操作成功!');location.href='
    Admin_News.jsp';</SCRIPT>");
}
```

任务 2：实现修改操作

(1) 打开“com.news.article”包中 ArticleManage 类，在修改新闻信息的 saveEdit(HttpServletRequest request)方法中添加如程序 8-30 所示的代码。

此方法的功能是修改一条新闻，在调用函数前要先选定想要修改的新闻 ID，如果没有选定，程序会抛出异常，并给出操作者相应的出错提示。然后会对“新闻名称”,“新闻分类”等进行非空判断和唯一性判断，如果为空时抛出异常，也会给出相应的提示。如果修改成功，则数据库的新闻表里对应的那条记录信息就更新了。

【程序 8-30】 saveEdit(HttpServletRequest request)方法

Example Source Code:

```
public static void saveEdit(HttpServletRequest request) throws Exception {
    int ID;
    try {
        ID=ParamUtil.getInt(request, "ID");
    } catch (Exception e) {
```



```
        throw new Exception("请您选择 ID.");
    }

    String sql="update Article set ArticleTitle=?, ArticleInfo=?, ArticleContent=?, ArticleKey
    =?, ArticleAuthor=?, ArticleFrom=?, ArticlePicture=?, MenuID=?, IsHead=?, IsImg=?,
    IsHot=?, SpecialID=?, AdminName=?where ID="+ID;

    DBConnect dbc=new DBConnect();
    dbc.prepareStatement(sql);

    dbc.setString(1, new String(ParamUtil.getString(request, ArticleTitle", "").
      getBytes("ISO-8859-1"), "GBK"));
    dbc.setString(2, new String(ParamUtil.getString(request, "ArticleInfo", "").
      getBytes("ISO-8859-1"), "GBK"));
    dbc.setString(3, new String(ParamUtil.getString(request, "ArticleContent", "").
      getBytes("ISO-8859-1"), "GBK"));
    dbc.setString(4, new String(ParamUtil.getString(request, "ArticleKey", "").
      getBytes("ISO-8859-1"), "GBK"));
    dbc.setString(5, new String(ParamUtil.getString(request, "ArticleAuthor", "").
      getBytes("ISO-8859-1"), "GBK"));
    dbc.setString(6, new String(ParamUtil.getString(request, "ArticleFrom", "").
      getBytes("ISO-8859-1"), "GBK"));
    dbc.setString(7, ParamUtil.getString(request, "ArticlePicture", ""));
    dbc.setInt(8, ParamUtil.getInt(request, "MenuID", 0));
    dbc.setInt(9, ParamUtil.getInt(request, "IsHead", 0));
    dbc.setInt(10, ParamUtil.getInt(request, "IsImg", 0));
    dbc.setInt(11, ParamUtil.getInt(request, "IsHot", 0));
    dbc.setInt(12, ParamUtil.getInt(request, "SpecialID", 0));
    dbc.setString(13, new String(ParamUtil.getString(request, "AdminName", "").
      getBytes("ISO-8859-1"), "GBK"));

    dbc.executeUpdate();

    dbc.close();
}
```

（2）打开第 7 章中建立的 Admin_Articles.jsp 文件，完善对应的修改界面的设计部分，所需代码如程序 8-31 所示。此处代码用于产生修改界面，即用于呈现所修改数据的表单页面，而并不是将数据进行修改处理的业务程序。

【程序 8-31】 新闻管理模块表现层的实现

Example Source Code:

```
⋮
    } else if (Action.equals("edit")) {
        Article theNew=(Article) ArticleManager.getArticle(ID);
%>
<form name="AddNews" method="POST"
        action="?Action=saveedit&ID=<%=theNew.getId()%>"
        onSubmit="return CheckNews();">
    <table>
```



```
<tr>
    <td align="right">所属大类：</td>
    <td colspan="2">
    <select name="ClassID" size="1" id="ClassID"
onChange="ChangeLocation(document.AddNews.ClassID.options[document.AddNews.ClassID.
selectedIndex].value)">
<%
    int i;
    Vector ClassVector=MenuManage.getMenus(0);
    if (ClassVector.size() >0) {
        out.println("<option value=\"\">请选择大类</option>");
        for (i=0; i<ClassVector.size(); i++) {
            Menu tempClass= (Menu) ClassVector.get(i);
            out.print("<option value=\""+tempClass.getMenuID()+"\"");
            if (tempClass.getMenuID()==theNew.getMenuID())
                out.print(" selected");
            out.println(">"+tempClass.getMenuName()+"</option>");
        }
    } else {
        out.println("<option value=\"\" selected>暂无大类</option>");
    }
%>
    </select>
    <span class="info1"> * 请选择新闻的类别</span>
    <select name="SpecialID" size="1" id="SpecialID"
        onChange="ChangeLocation(document.AddNews.BigClassID.options[document.
          AddNews.BigClassID.selectedIndex].value)">
        <option selected value="">不属于任何专题</option>
    </select>
    </td>
</tr>
<tr>
    <td align="right">标题：</td>
    <td colspan="2">
    <input name="NewsTitle" type="text" id="NewsTitle" value="<%=theNew.getTitle ()%>"
    size="40" maxlength="60">
    </td>
</tr>
<tr>
    <td align="right">发布人：</td>
    <td width="120">
    <input name="NewsAuthor" type="text" id="NewsAuthor" value="<%=theNew.getAuthor
    ()%>" size="10" maxlength="20">
    </td>
    <td>出处： <input name="NewsFrom" type="text" id="NewsFrom"
        value="<%=theNew.getFrom()%>" size="10" maxlength="60" /></td>
</tr>
<tr>
    <td align="right">关键字：</td>
    <td colspan="2"><input name="NewsKey" type="text" id="NewsKey"
        value="<%=theNew.getKey()%>" size="40" maxlength="30"></td>
</tr>
<tr>
```



```
        <td align="right">图片新闻：</td>
        <td><label><input type="radio" name="IsImg" value="1"
            onClick="ShowPicture(true)">是</label><label><input
            name="IsImg" type="radio" value="0" checked
            onClick="ShowPicture(false)">否</label></td>
        <td>
        <div id="p" style="display: none">  <input
            name="NewsPicture" type="text" id="NewsPicture" size=25
            maxlength="80" style="display: " value=""><input
            name="UpImg" type="button" id="UpImg" onClick="getPicture()"
            value="新闻图片>>"> <span class="info1">(130*80)</span></div>
        </td>
    </tr>
    <tr>
        <td align="right">头条新闻：</td>
        <td><label><input type="radio" name="IsHead" value="1"
            onClick="ShowHeadPicture(true)">是</label><label><input
            name="IsHead" type="radio" value="0" checked
            onClick="ShowHeadPicture(false)">否</label></td>
        <td>
        <div id="p1" style="display: none">  <input
            name="HeadPicture" type="text" id="HeadPicture" size=25
            maxlength="80" style="display: " value=""><input
            name="UpImg" type="button" id="UpImg" onClick="getHeadPicture()"
            value="头条图片>>"> <span class="info1">(250*140)</span></div>
        </td>
    </tr>
    <tr>
        <td align="right">滚动新闻：</td>
        <td colspan="2"><label><input name="IsHot" type="radio"
            value="1">是</label><label><input name="IsHot" type="radio"
            value="0" checked>否</label></td>
    </tr>
    <tr>
        <td align="right">新闻摘要：</td>
        <td colspan="2"><textarea name="NewsInfo" cols="40" rows="3"
            id="NewsInfo"><%=theNew.getInfo()%></textarea><span class="info1">
              *120字以内</span></td>
    </tr>
    <tr>
        <td colspan="3">
        <FCK:editor id="NewsContent"
        basePath="/FCKeditor/" width="600" height="400"
        skinPath="/FCKeditor/editor/skins/silver/" defaultLanguage="zh-cn"
        tabSpaces="8"

        imageBrowserURL="/FCKeditor/editor/filemanager/browser/default/browser.
          html?Type=Image&Connector=connectors/jsp/connector"

        linkBrowserURL="/FCKeditor/editor/filemanager/browser/default/browser.html?
          Connector=connectors/jsp/connector"

        flashBrowserURL="/FCKeditor/editor/filemanager/browser/default/browser.html?
```



```
        Type=Flash&Connector=connectors/jsp/connector"

      imageUploadURL="/FCKeditor/editor/filemanager/upload/simpleuploader?Type
        =Image"
          linkUploadURL="/FCKeditor/editor/filemanager/upload/simpleuploader?
            Type=File"

      flashUploadURL="/FCKeditor/editor/filemanager/upload/simpleuploader?Type
        =Flash">
          <%=theNew.getContent()%>
      </FCK:editor>
      </td>
    </tr>
    <tr>
      <td colspan=3 align="center"><input name="B1" type="button"
          value="返回>>" onClick="location.href='ListNews.jsp'"><input
          name="Action" type="hidden" id="Action" value="Add"><input
          name="AdminName" type="hidden" id="AdminName"
          value="<%=new User(Integer.parseInt(Session.getAttribute("AdminID").
            toString())).getName()%>">
       <input name="B2" type="submit" id="B2" value="增加>>"></td>
    </tr>
</table>
    </form>
<%
```

(3)打开第7章中建立的Admin_Articles.jsp文件,完善对应的处理修改操作的部分,所需代码如程序8-32所示。此处代码是处理修改操作的程序,即用于进行数据处理的业务层,将调用程序8-30所示的ArticleManage类的saveEdit(HttpServletRequest request)方法,把从修改数据的表单中接收的数据写入数据库中。

【程序8-32】 新闻管理模块表现层的实现

Example Source Code:

```
 ⋮
    } else if (Action.equals("saveedit")) {
        ArticleManager.saveEdit(request);
        out.println("saveedit操作成功!");
        out.println("<SCRIPT LANGUAGE='JavaScript'>alert('del操作成功!');
          location.href='Admin_News.jsp';</SCRIPT>");
```

任务3:实现删除操作

(1)打开"com.news.article"包中的ArticleManage类,在删除新闻信息的delArticles(HttpServletRequest request)方法中添加如程序8-33所示的代码。

程序8-33是删除一条新闻的方法,调用此方法要事先选择欲删除新闻的ID,如果没有选定,程序要抛出异常,并给出操作者相应的出错提示,然后执行删除操作。如果删除成功,则数据库中新闻表里对应的那条记录信息就不存在了。

【程序8-33】 delArticles(HttpServletRequest request)方法

Example Source Code:



```
public static void delArticles(HttpServletRequest request) throws Exception {
    int ArticleID;
    try {
        ArticleID= ParamUtil.getInt(request, "ID");
    } catch (Exception e) {
        throw new Exception("请您选择新闻的 ID.");
    }
    DBConnect dbc= new DBConnect("delete from Article where id= "+ ArticleID);
    dbc.executeUpdate( );
    dbc.close( );
}
```

(2) 打开第 7 章中建立的 Admin_Articles.jsp 文件，完善对应的处理删除操作的部分，所需代码如程序 8-34 所示。此处代码是处理删除操作的程序，即用于进行数据处理的业务层，将调用程序 8-33 所示的 ArticleManage 类的 delArticles (HttpServletRequest request) 方法，直接从数据库中将所选数据删除。

【程序 8-34】 新闻管理模块表现层的实现

Example Source Code:

```
} else if (Action.equals("del")) {
        ArticleManager.delArticles(request);
        out.println("<SCRIPT LANGUAGE= 'JavaScript'> alert('del 操作成功！');location.href
        = 'Admin_News.jsp';</SCRIPT> ");
```

8.4.3　相关知识拓展

在线编辑器对于新闻产品之类的网站是很重要的，它使网站管理员可以像在 Word 中编辑文字一样对发布的内容进行编辑。在"新闻类别网站"的新闻发布中使用了 FCKeditor 在线编辑器来编辑将要发布的内容，具体调用方式采用的是 FCKeditor 标签实现的。

1. FCKeditor 简介

FCKeditor 是一个专门使用在网页上属于开放源代码的所见即所得式文字编辑器。它注重于轻量化，不需要太复杂的安装步骤即可使用。它可和 PHP、JavaScript、ASP、ASP.NET、ColdFusion、Java 以及 ABAP 等不同的编程语言相结合。FCKeditor 名称中的 FCK 是这个编辑器的作者名字 Frederico Caldeira Knabben 的缩写。

FCKeditor 控件可以到官方网站 http://www.fckeditor.net 中下载，本例主要用到 FCKeditor_2.6.3.zip、FCKeditor-java-demo-2.4.1.zip、FCKeditor-java-2.4.1-bin.zip 里面的一些内容，读者可以自行下载。由于版本更新快，针对不同版本的具体设置可参考网上资料。

2. 使用 FCKeditor 的方式

在页面中使用 FCKeditor 有两种方式。方式一是 JavaScript，方式二是利用标签使用 FCKeditor。

(1) JavaScript 方式



① 在 JavaScript 中直接使用 FCKeditor,如程序 8-35 所示。

【程序 8-35】 method1. html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
    <script type="text/javascript" src="fckeditor/fckeditor.js"></script>
</head>
<body>
    方法一:通过 JavaScript 来实现的在线编辑器导入<br>
    <script type="text/javascript">
    var oFCKeditor=new FCKeditor('FCKeditor1');
    oFCKeditor.BasePath="fckeditor/";
    //oFCKeditor.BasePath="/FCKEditTest/fckeditor/";
    oFCKeditor.Width="50% ";
    oFCKeditor.Height="400";
    oFCKeditor.Value="initial value";
    //oFCKeditor.ToolbarSet="Basic";
    //默认是 default
    oFCKeditor.ToolbarSet="Default";
    oFCKeditor.Create();
    </script>
    <hr>
</body>
</html>
```

效果如图 8-5 所示。

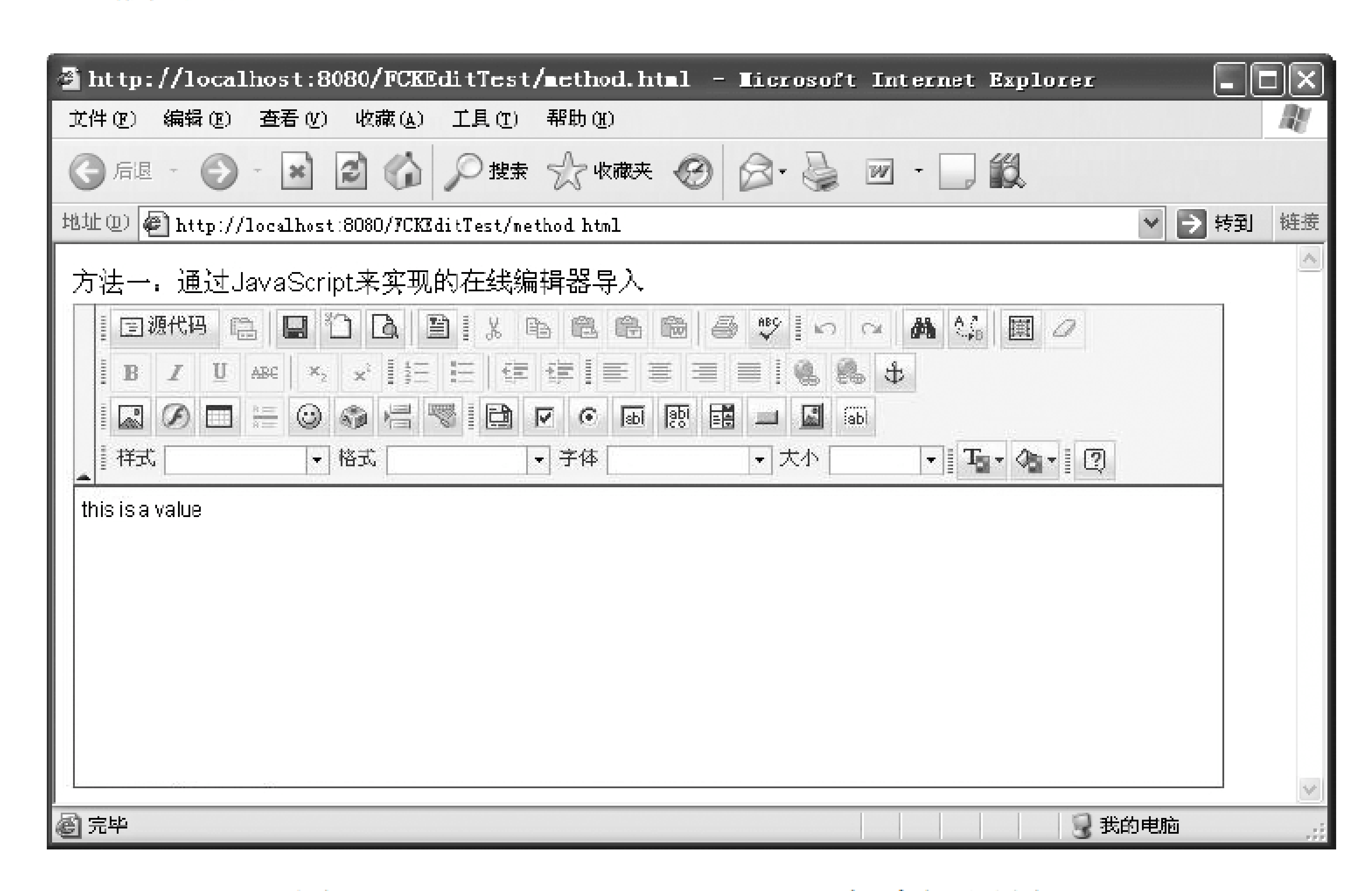


图 8-5 FCKeditor JavaScript 方式调用界面

② 通过 Textarea 使用 FCKeditor,如程序 8-36 所示。

【程序 8-36】 method2. html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<script type="text/javascript" src="fckeditor/fckeditor.js"></script>
```



```
<script type="text/javascript">
window.onload=function(){
    var oFCKeditor=new FCKeditor('MyTextarea');
    oFCKeditor.BasePath="fckeditor/";
    //oFCKeditor.BasePath ="/FCKEditTest/fckeditor/";
    oFCKeditor.ReplaceTextarea();
}
</script>
</head>
<body>
    方法二：通过 Textarea 来实现的在线编辑器导入<br>
    <textarea rows="4" cols="60" name="MyTextarea">this is a value</textarea>
</body>
</html>
```

效果如图 8-6 所示。

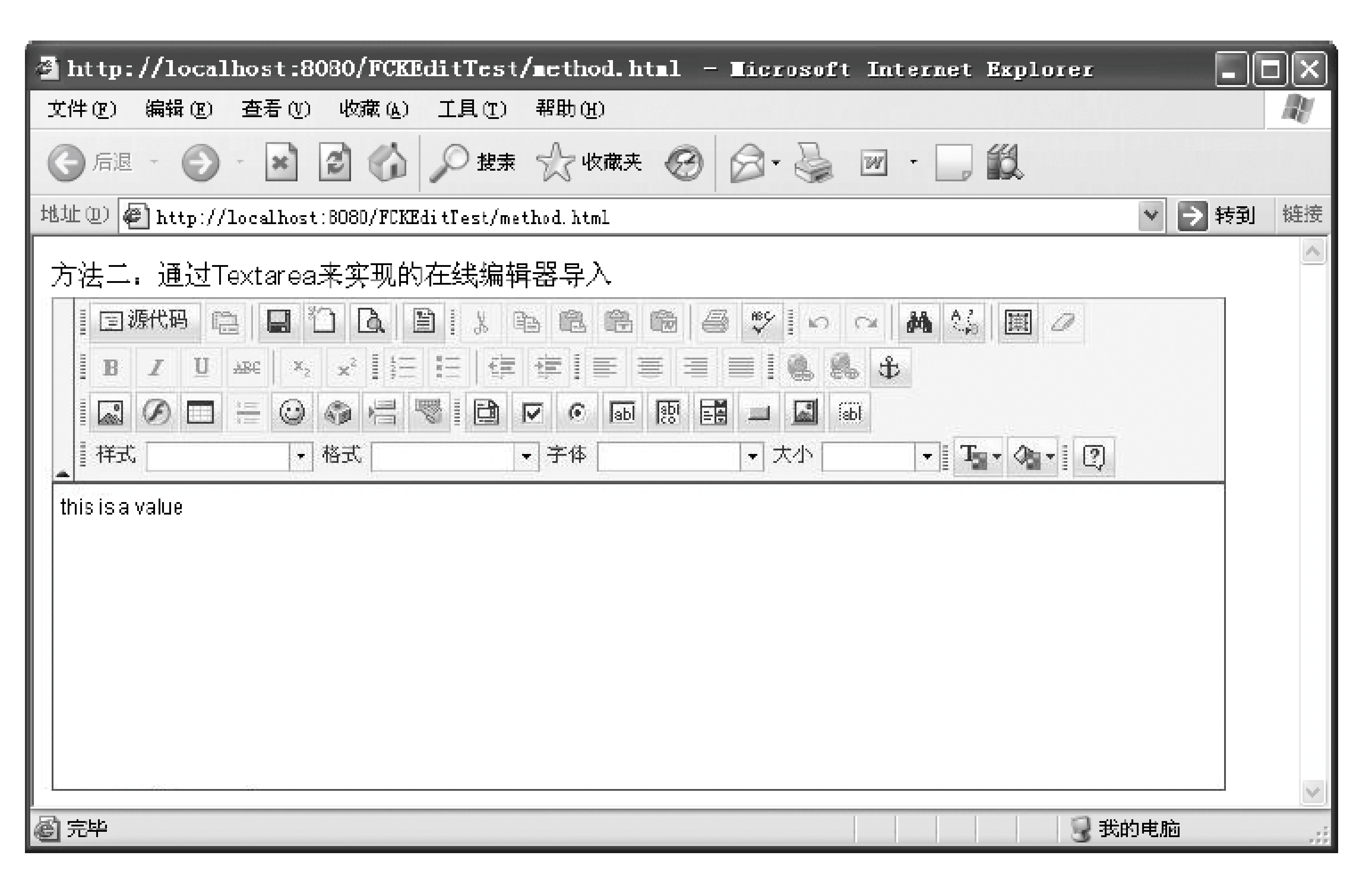


图 8-6　FCKeditorTextarea 方式调用界面

（2）利用标签使用 FCKeditor

在使用标签之前，需要搭建环境。将 fckeditor-java-2.4.1-bin.zip 包中的 fckeditor-java-core-2.4.1.jar、commons-fileupload-1.2.1.jar、commons-io-1.3.2.jar、slf4j-api-1.5.2.jar 以及包 fckeditor-java-demo-2.4.1.zip 中的 slf4j-simple-1.5.2.jar 复制到 lib 目录下，并构建环境。

构建完后，就可以在 JSP 页面中调用 FCKeditor 了，如程序 8-37 所示。在“新闻类别网站”的新闻发布中就是采用这种方式调用 FCKeditor 在线编辑器的。

【程序 8-37】　method_jsp.jsp

```
<%@page language="java" import="java.util.*" pageEncoding="utf-8"%>
<%@taglib uri="http://java.fckeditor.net" prefix="FCK"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>method_jsp.jsp</title>
</head>
<body>
```



早些版本必须将 fckeditor 的 Value 属性必须指定对应的值,且值不能为空。而此处版本是 2.6.3,该问题已经解决。

```
<FCK:editor instanceName="myEditor" basePath="/fckeditor"></FCK:editor>
</body>
</html>
```

8.4.4 经验总结

在基于 Java 的编程中,经常会碰到汉字处理及显示出现问题,比如一大堆乱码或问号。这是因为 Java 中默认的编码方式是 unicode,而中国人通常使用的文件和 DB 都是基于 GB2312 或者 BIG5 等编码,故会出现此问题。

(1) 在网页中输出中文

Java 在网络传输中使用的编码是“ISO-8859-1”,故在输出时需要进行转化,如:

```
String str="中文";
str=new String(str.getBytes("GB2312"),"8859_1");
```

假如在编译程序时,使用的编码是“GB2312”,而且是在中文平台上运行此程序,不会出现此问题,不需要转化。

(2) 从参数中读取中文

这正好与在网页中输出相反。如:

```
str=new String(str.getBytes("8859_1"),"GB2312");
```

(3) 操作 DB 中的中文问题

一个较简单的方法是:在“控制面板”中,把“区域”设置为“英语(美国)”。假如还会出现乱码,还可进行如下设置:

```
取中文时: str=new String(str.getBytes("GB2312"));
向 DB 中输入中文: str=new String(str.getBytes("ISO-8859-1"));
```

(4) 解决 JSP 中的中文问题

假如中文不能正常显示,可进行下面的转换,如:

```
name=new String(name.getBytes("ISO-8859-1"),"GBK");
```

就不会出现中文问题了。

8.5 新闻的查询

8.5.1 任务描述

利用 Eclipse 实现“新闻类别网站”的新闻查询模块。



8.5.2　任务实现

具体操作步骤如下：

(1) 打开"com. news. article"包中的 ArticleManage 类，在获取指定 ArticleID 新闻的 getArticle(int ArticleID)方法中添加如程序 8-38 所示的代码。

此方法的功能是获取指定 ArticleID 的新闻。如果有异常发生时，将产生 ArticleNotFoundException 异常，如果执行成功，则会返回类型为 Article 的指定 ID 的新闻。

【程序 8-38】 getArticle(int ArticleID) 方法

Example Source Code:

```
public static Article getArticle(int ArticleID) throws ArticleNotFoundException {

    DBConnect dbc;
    try {
        dbc=new DBConnect("select * from Article where id="+ ArticleID);
        ResultSet rs=dbc.executeQuery();
        if (!rs.next())
            throw new Exception("对不起,没有发现此新闻!");
        // rs.next();
        Article tempArticle=new Article();

        tempArticle.setId(rs.getInt(1));
        tempArticle.setTitle(rs.getString(2));
        tempArticle.setInfo(rs.getString(3));
        tempArticle.setContent(rs.getString(4));
        tempArticle.setKey(rs.getString(5));
        tempArticle.setAuthor(rs.getString(6));
        tempArticle.setFrom(rs.getString(7));
        tempArticle.setAddTime(rs.getString(8));
        tempArticle.setPicture(rs.getString(9));
        tempArticle.setMenuID(rs.getInt(10));
        tempArticle.setIsHead(rs.getInt(11));
        tempArticle.setIsImg(rs.getInt(12));
        tempArticle.setIsHot(rs.getInt(13));
        tempArticle.setSpecialID(rs.getInt(14));
        tempArticle.setAdminName(rs.getString(15));
        tempArticle.setHit(rs.getInt(16));

        return tempArticle;
    } catch (Exception e) {
        e.printStackTrace();
        throw new ArticleNotFoundException();
    }
}
```

(2) 打开"com. news. article"包中的 ArticleManage 类，在获取指定 MenuID 和类别新闻的 getArticles(int MenuID,String type)方法中添加如程序 8-39 所示的代码。



此方法的功能是获取父 ID 为 MenuID 的所有新闻，即返回该新闻分类下的所有新闻。在调用方法时会根据 type 的不同形成不同的结果，从而使方法的功能更加强大。如果有异常发生时，将产生 ArticleNotFoundException 异常；如果执行成功，则会返回指定新闻分类下的指定分类下的所有新闻。

【程序 8-39】 getArticles(int MenuID,String type) 方法

Example Source Code：

```
public static Vector getArticles(int MenuID,String type) throws ArticleNotFound
  Exception {
    DBConnect dbc;
    String sql="select * from Article where MenuID="+MenuID;
    try {
        if (type=="hot"){
            sql+=" order by Hit desc,ArticleTime DESC";
        }else if (type=="new"){
            sql+=" order by ArticleTime DESC";
        }else if (type=="HH"){
            sql+=" order by Hit desc,ArticleTime DESC";
        }else
            sql+=" order by ID desc,ArticleTime DESC";

        dbc=new DBConnect(sql);
        ResultSet rs=dbc.executeQuery();
        Vector articleVector=new Vector();
        while (rs.next()) {
            Article tempArticle=new Article();

            tempArticle.setId(rs.getInt(1));
            tempArticle.setTitle(rs.getString(2));
            tempArticle.setInfo(rs.getString(3));
            tempArticle.setContent(rs.getString(4));
            tempArticle.setKey(rs.getString(5));
            tempArticle.setAuthor(rs.getString(6));
            tempArticle.setFrom(rs.getString(7));
            tempArticle.setAddTime(rs.getString(8));
            tempArticle.setPicture(rs.getString(9));
            tempArticle.setMenuID(rs.getInt(10));
            tempArticle.setIsHead(rs.getInt(11));
            tempArticle.setIsImg(rs.getInt(12));
            tempArticle.setIsHot(rs.getInt(13));
            tempArticle.setSpecialID(rs.getInt(14));
            tempArticle.setAdminName(rs.getString(15));
            tempArticle.setHit(rs.getInt(16));

            articleVector.add(tempArticle);

        }
        dbc.close();
        return articleVector;
```



```
        } catch (Exception e) {
            e.printStackTrace();
            throw new ArticleNotFoundException();
        }
    }
```

（3）打开“com. news. article”包中的 ArticleManage 类，在获取所有新闻的getArticles( )方法中添加如程序 8-40 所示的代码。

此方法的功能是获取所有新闻，在调用方法时会对“新闻”按照 ID 进行排序。如果有异常发生时，将产生 ArticleNotFoundException 异常；如果执行成功，则会返回由所有新闻组成的 Vector 类型变量。

【程序 8-40】 getArticles( ) 方法

Example Source Code：

```
public static Vector getArticles() throws ArticleNotFoundException {
    DBConnect dbc;
    try {
        dbc=new DBConnect("SELECT * FROM Article ORDER BY ID DESC");
        ResultSet rs=dbc.executeQuery();
        Vector articleVector=new Vector();
        while (rs.next()) {
            Article tempArticle=new Article();

            tempArticle.setId(rs.getInt(1));
            tempArticle.setTitle(rs.getString(2));
            tempArticle.setInfo(rs.getString(3));
            tempArticle.setContent(rs.getString(4));
            tempArticle.setKey(rs.getString(5));
            tempArticle.setAuthor(rs.getString(6));
            tempArticle.setFrom(rs.getString(7));
            tempArticle.setAddTime(rs.getString(8));
            tempArticle.setPicture(rs.getString(9));
            tempArticle.setMenuID(rs.getInt(10));
            tempArticle.setIsHead(rs.getInt(11));
            tempArticle.setIsImg(rs.getInt(12));
            tempArticle.setIsHot(rs.getInt(13));
            tempArticle.setSpecialID(rs.getInt(14));
            tempArticle.setAdminName(rs.getString(15));
            tempArticle.setHit(rs.getInt(16));

            articleVector.add(tempArticle);
        }
        dbc.close();
        return articleVector;
    } catch (Exception e) {
        e.printStackTrace();
        throw new ArticleNotFoundException();
    }
}
```



(4) 打开"com. news. article"包中的 ArticleManage 类，在用于获取 ID 为 MenuID 的所有新闻的 getArticles (int MenuID)方法中添加如程序 8-41 所示的代码。

此方法的功能是获取 ID 为 MenuID 的所有新闻，即返回该新闻分类下的所有新闻。如果有异常发生时，将产生 Exception 异常；如果执行成功，则会返回指定新闻分类下的所有新闻。

【程序 8-41】 getArticles (int MenuID) 方法

Example Source Code:

```
public static Vector getArticles(int MenuID) throws Exception {
    try {
        Vector articleVector=new Vector( );
        articleVector=gArticles(MenuID);
        Vector menuVector=new Vector( );
        menuVector=MenuManage.getMenus(MenuID);
        int i;
        if (menuVector.size( )>0) {
            for (i=0; i<menuVector.size( ); i++) {
                Menu tempMenu = (Menu)menuVector.get(i);
                if (getArticles(tempMenu.getMenuID( )).size( )>0) {
                    int j;
                    for (j=0; j<getArticles(tempMenu.getMenuID( )).size( ); j++) {
                        Article theNews= (Article) getArticles(
                                tempMenu.getMenuID( )).get(j);
                        articleVector.add(theNews);
                }
                }
            }
        }
        return articleVector;
    } catch (Exception e) {
        e.printStackTrace( );
        throw new Exception("请您选择新闻的 ID.");
    }
}
```

(5) 打开第 7 章中建立的 Admin_Articles. jsp 文件，完善对应的处理删除操作的部分，所需代码如程序 8-42 所示。此处代码是用于处理新查询操作，即用于进行数据处理的业务层，将调用 ArticleManage 类的相应方法，直接从数据库中查找相应的数据，并显示出来。

【程序 8-42】 新闻管理模块表现层的实现

Example Source Code:

```
⋮
if (Action.equals("list")) {
    Vector news;
    if (ClassID==0)
        news=ArticleManager.getArticles( );
    else
        news=ArticleManager.getArticles(ClassID);
%>
```






```
<form name="searchuser" method="POST" action=""><select
    name="ClassID" id="ClassID">
    <option value="0">请选择分类</option>
    <option value="1">1</option>
    <option value="2">2</option>
</select><input type="Title" name="Title" size="30"><select name="Row"
    id="Row">
    <option value="1">按标题</option>
    <option value="2">按内容</option>
    <option value="3">关键字</option>
</select><input type="submit" value="查询" name="title"></form>
<table>
    <tr>
        <th>ID</th>
        <th>标题</th>
        <th>分类</th>
        <th>作者</th>
        <th>添加时间</th>
        <th>操作</th>
    </tr>
    <%
        int i;
        int orders=ParamUtil.getInt(request, "orders", 7);
        int usersNum=news.size();
        int Page=ParamUtil.getInt(request, "Page", 1);
        for (i=(Page-1) * 5; i<Page * 5; i++) {
            if (i>=usersNum)
                break;
            Article theNews=(Article) news.get(i);
    %>
    <tr>
        <td><input name="ID" type="checkbox" id="ID"
            value="<%=theNews.getId()%>" /></td>
        <td><a href="Content.jsp?ID=<%=theNews.getId()%>"><%=Format.CutStr
          (theNews.getTitle(), 25)%></a></td>
        <td><%=new Menu(theNews.getMenuID()).getMenuName()%></td>
        <td><%=theNews.getAuthor()%></td>
        <td><%=theNews.getAddTime()%></td>
        <td><a href="?Action=edit&ID=<%=theNews.getId()%>">修改</a> |<a
            href="?Action=del&ID=<%=theNews.getId()%>">删除</a></td>
    </tr>
    <%
        }
    %>
    <tr>
        <td colspan="6" align="center">
        <%
            int totalPages=usersNum/5+1;
            if (Page==0)
                Page=1;
            out.println("共 "+totalPages+" 页第 "+Page+" 页 ");
```



```
            if (Page==1 || totalPages==1)
                out.println("<font color=gray>首页  前页</font>");
            else
                out.println("<a href=?Page=1&orders="+orders
                        +"&txtitle="
                        +ParamUtil.getString(request, "txtitle", "")
                        +">首页</a>  <a href=?Page="+(Page-1)
                        +"&orders="+orders+"&txtitle="
                        +ParamUtil.getString(request, "txtitle", "")
                        +">前页</a>");

            if (Page==totalPages || totalPages==1)
                out.println(" <font color=gray>后页  尾页</font>");
            else
                out.println(" <a href=?Page="+(Page+1)+"&orders="
                        +orders+"&txtitle="
                        +ParamUtil.getString(request, "txtitle", "")
                        +">后页</a>  <a href=?Page="+totalPages
                        +"&orders="+orders+"&txtitle="
                        +ParamUtil.getString(request, "txtitle", "")
                        +">尾页</a>");
        %>
        </td>
    </tr>
</table>
<%
```

8.5.3 相关知识拓展

分页是一种常用的页面数据显示技术，分页是通过减少页面数据处理量来提高系统的性能。分页是做 Web 开发必须掌握的一个小技术，但分页却是复杂的，并不是分页技术有多复杂，而是有太多的重复代码，且这些代码都难以重用。所以重要的是把这些分页代码和业务逻辑分离，同时提供足够的方便性。下面总结一下分页的几种方案。

1. 数据岛

后台将数据封装到 XML 格式的文本串中，将该文本串传入前台，由 JS 解析出特定页的数据，并显示出来。

这种方式的优点是：数据传输方便，且 XML 本身的特点（这里主要是扩展性，可以根据自己的需要自定义 XML 的格式）可以发挥出来；分页功能在一定程度上可复用（JS 接收 XML，与业务逻辑相分离）；一次传输多页甚至全部数据（根据数据量权衡），可以减少与数据库的交互。

这种方式的缺点是：实时性差，第一次访问速度欠佳（因为第一次一般会取出比较多的数据）；过于依赖 JS，虽然简化了数据的显示，但是后续工作的难度却加大了，如取值、验证，另外，JS 是一种结构化的弱类型的语言，它的灵活性直接使调试的难度加大，它的大量使用导致程序难以维护。



2. 简化后的数据岛

这种方法直接使用 DHTML 的特性。TABLE 的 dataSrc 属性和 nextPage 等翻页方法，相对简单地解决了分页问题。

3. OO(Object Orient)方法

这种方法是定义一个分页接口，在其中定义一系列的标准操作，在一个抽象类中实现该接口，完成分页逻辑，同时对外提供两个抽象方法(Hook method)：

```
int getTotalCount( );                          //取得所有记录的数目
List getPageDate(int startRow, int rowCount)     //取得从 startRow 开始的 rowCount 条记录
```

使用时，只需覆盖这两个方法，就可以完成与业务逻辑的对接(Template Method)。这种方法的优点是：面向对象，易复用、易扩展、易理解；与具体的业务、框架、数据源分离；前台(JSP)页面上的代码也可复用，基本不变；相对第一种方法而言，取值、验证简单了。这种方法的缺点是：每显示一页都需要访问数据源，对数据没有“缓存”功能，但可用数据库连接池(和 Session)弥补。

最后，“新闻类别网站”作为管理信息系统，必然会涉及大量数据在前台页面上的列表显示，这就要求系统具有分页显示数据信息的功能。在此总结一下完成分页功能的方法，即在 Bean 中实现分页功能和在 JSP 页面中实现分页功能。其实现原理都是在数据库中检索出要显示的记录，根据所设定的每页记录数算出总页数，在页面上显示时，根据传回的页码值决定当前要显示的页的记录。

另一种分页的思路是将查询结果缓存在 HttpSession 或有状态 Bean 中，翻页的时候从缓存中取出一页数据来显示，这种方法有两个主要的缺点：一是用户可能看到的是过期数据；二是如果数据量非常大，则第一次查询遍历结果集会耗费很长时间，并且缓存的数据也会占用大量内存，效率明显下降。

还有一种思路是每次翻页的时候只从数据库里检索页面大小的块区的数据。这样虽然每次翻页都需要查询数据库，但查询出的记录数很少，网络传输数据量不大。

8.5.4 经验总结

在 Web 程序中查询会被频繁使用，但查询的实现细节却是编程过程中比较麻烦的事情。大多分页显示的查询操作都同时需要处理复杂的多重查询条件，SQL 语句需要动态拼接组成，再加上分页需要的记录定位、总记录条数查询以及查询结果的遍历、封装和显示，程序会变得很复杂并且难以理解。

一般来说只要弄清楚条件的构成、查询和显示的问题，就不难实现查询。在实现查询时按照接受查询参数、形成记录集、数据显示和显示新的参数来实现就可以了。不过需要注意参数的传递是通过两种方式实现的，既可以通过表单传递，也可以通过查询字符串传递。

8.6 实　训

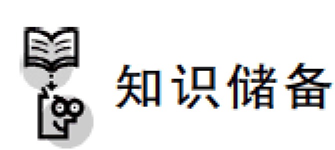
根据本章所述内容，完成“新闻类别网站”对数据库操作的编程任务。



第9章 测试网站

网页制作完成后，在正式上传到服务器之前，还有一项很重要的工作要做，那就是网站的测试。网站测试的目的是尽可能多地发现浏览器端和服务器端程序中的错误并及时加以修正，以保证网站的质量。网站测试主要包括功能测试、性能测试、兼容性测试、可用性测试和安全性测试等。

9.1 测试网站的功能



知识储备

网站的功能测试，是指检查网站是否实现了网站需求分析时定义的网站功能。网站功能测试时的测试任务具体包括内容测试、链接测试、表单测试等。

1. 内容测试

内容测试是测试网页上的内容是否被正常显示和准确表述。测试时需要检查网站所有的网页上是否有文本拼写错误、语法错误、图片显示错误等。

内容测试其实就像编辑审校工作一样，专门检验各个网页信息的正确性和准确性。首先，要检查网站上发布的信息是不是正确的、真实无误的，例如，需要检查网站上发布的各种通信信息是否正确，包括电话、通信地址、E-mail 信息等。其次，要查看网页信息的文字表述是否恰当，是否符合语法逻辑，是否存在拼写错误。

2. 链接测试

链接测试主要是检查网页中是否有错误的超级链接，包括文字链接、图片链接、服务器端程序链接等。Dreamweaver 软件提供了链接测试的功能，可以完成网站中页面链接的测试任务。

3. 表单测试

表单是用来向服务器提交信息的，它提供了一种允许用户从键盘输入或用鼠标选择信息的交互式界面，实现了网页中的数据获取功能，例如用户注册界面、登录界面等。表单测试就是测试表单中所提交的信息是否正确。例如，用户填写的密码与确认密码是否相同、填写的邮箱地址是否包括@、填写的出生日期与职业是否恰当，填写的所属省份与所在城市是否相匹配等。

9.1.1　任务描述

“新闻类别网站”的测试工作任务之一：完成网站的功能测试。

9.1.2　任务实现

“新闻类别网站”的功能测试分三个任务：

任务 1　实施内容测试
任务 2　实施链接测试
任务 3　实施表单测试

任务 1：实施内容测试

具体操作步骤如下：

（1）打开“新闻类别网站”主页，先检查主页中是否有内容错误，如图 9-1 所示。具体需要检查的有：图片或动画是否能正常显示，网页文字是否有语法错误和拼写错误，标题、日期、地址等信息是否正确等。如果发现存在错误内容，则通过修改页面代码、文本或者数据库中的数据，使其变为正确状态。



图 9-1　检查网站主页

（2）依次单击主页上的所有新闻链接，检查所有打开页面的新闻内容，如果有内容错误则进行修改。

（3）单击主页底部的“管理入口”，进入后台管理页面，如图 9-2 所示。后台管理功能包括系统管理、新闻管理、用户管理等。检查每一项后台功能对应的页面是否有错误，有则改之。



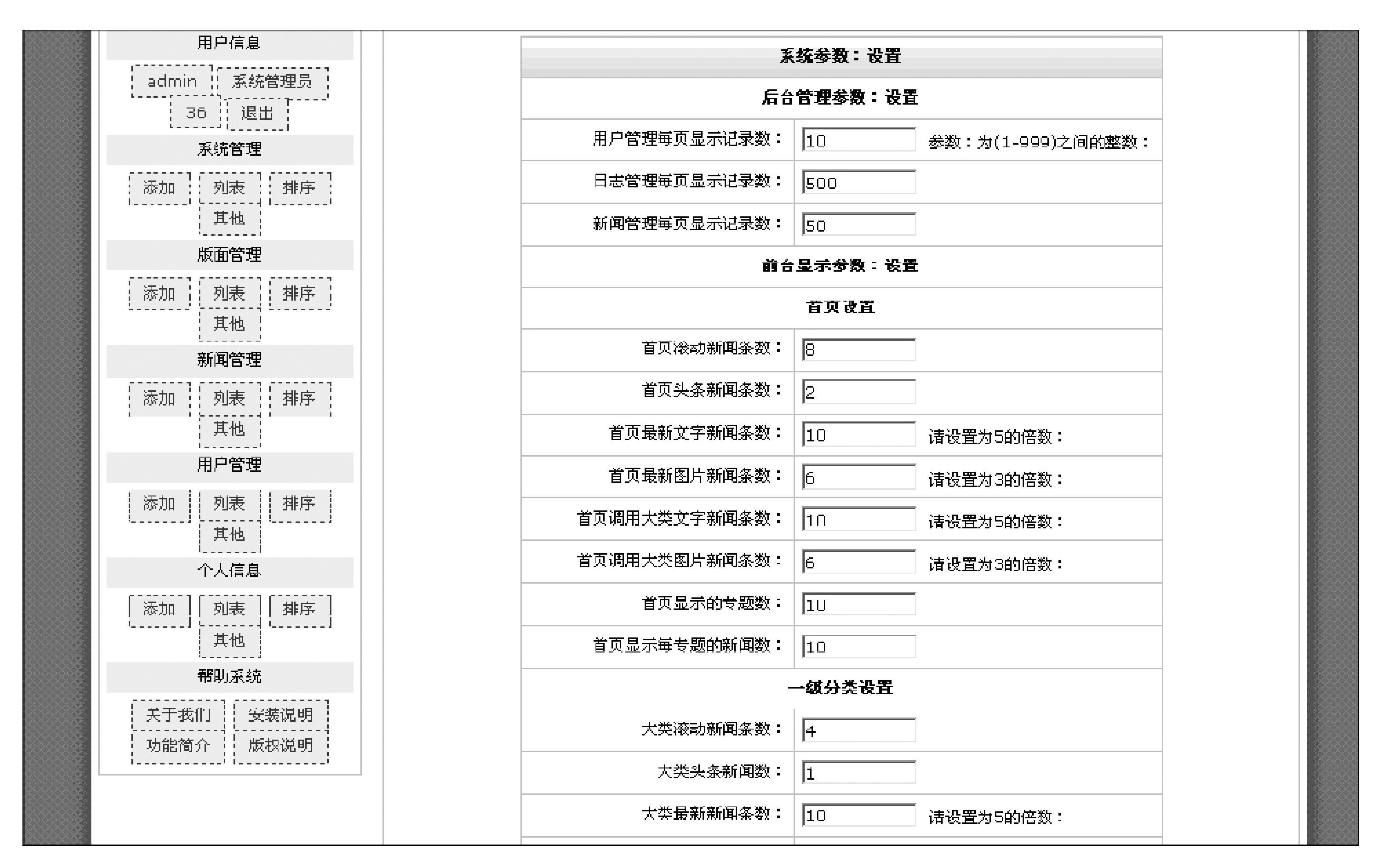


图 9-2　后台管理页面

任务 2：实施链接测试

具体操作步骤如下：

（1）选择需要检查链接的页面，选择 Dreamweaver 菜单中的【窗口】|【结果】命令，打开【结果】对话框，如图 9-3 所示。

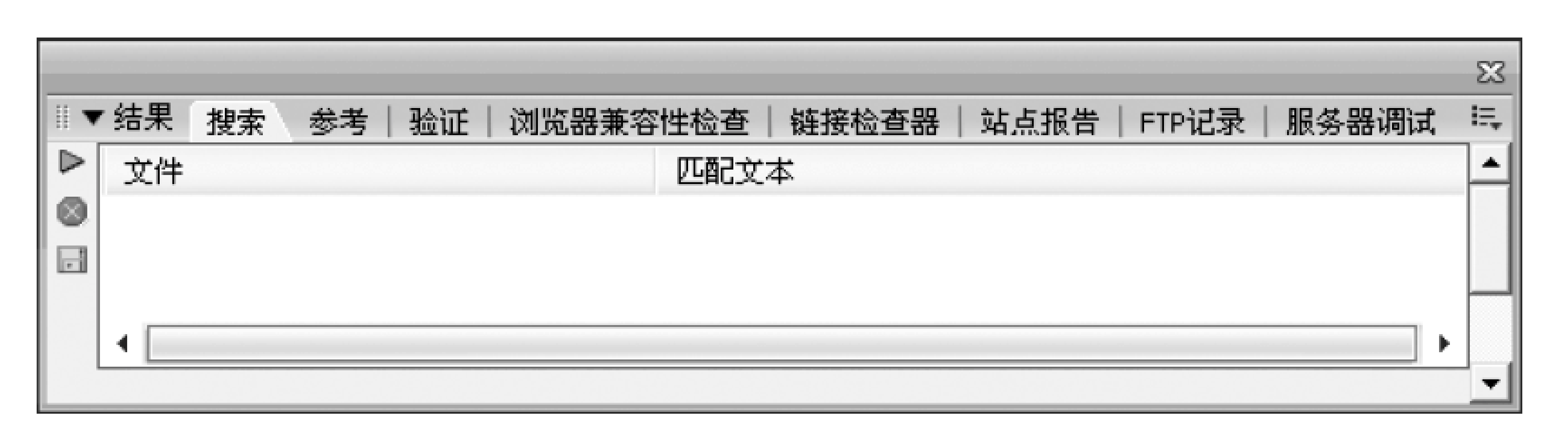


图 9-3　【结果】对话框

（2）在【结果】对话框中，单击【链接检查器】标签。在【显示】下拉列表框中选择要使用的链接检查方式，此处选择“断掉的链接”，如图 9-4 所示。



图 9-4　【链接检查器】对话框

小贴士

Dreamweaver 中提供的链接测试有：断掉的链接、外部链接和孤立文件。



（1）断掉的链接测试

断掉的链接测试是测试站点或文档中是否有断开的链接，即某一链接没有正确链接到它应该链接的页面。

（2）外部链接测试

外部链接测试是显示所有的外部链接，即链接到本网站外部的链接。

（3）孤立文件测试

孤立文件测试是测试站点中是否存在孤立的网页，即没有链接指向该页面。【孤立文件】选项只在【检查整个当前本地站点的链接】(此选项参见如图 9-5 所示)操作中才有效。

（3）单击【结果】面板左侧的绿色三角箭头按钮，在弹出的下拉菜单中选择“链接检测的范围”，如图 9-5 所示。

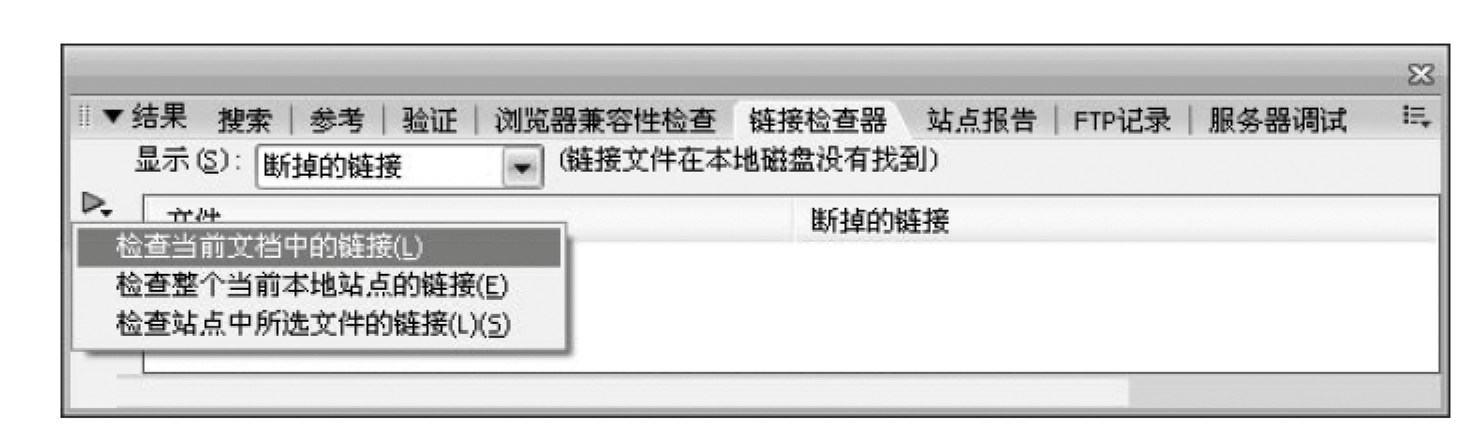


图 9-5　【检测范围】下拉菜单

小贴士

Dreamweaver 允许检测当前文档的链接、整个站点的链接或部分站点文件中的链接。

选择“检查当前文档中的链接”，则弹出显示当前文档中链接检查的报告单。

选择“检查整个当前本地站点的链接”，则对整个站点进行检查。

选择“检查站点中所选文件的链接”，则可以针对部分文档进行检查。

（4）在【显示】下拉列表框中选择“断掉的链接”，检查范围是“检查整个当前本地站点的链接”，则在面板中显示“断掉的链接”的测试结果，如图 9-6 所示。

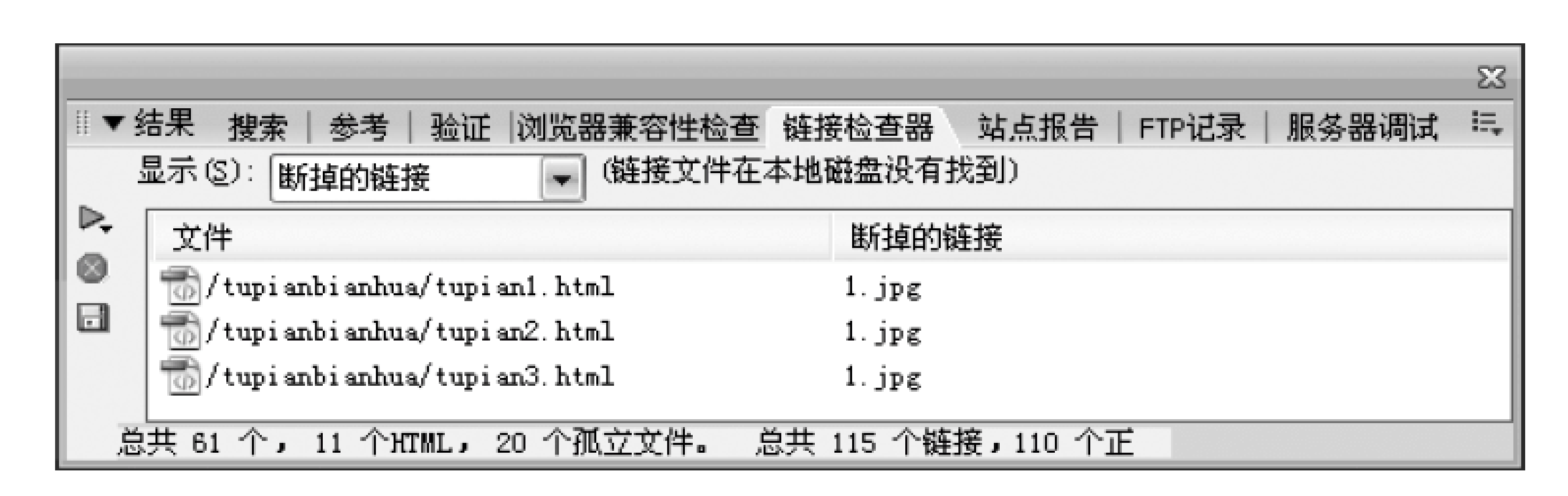


图 9-6　“断掉的链接”测试结果

如果想要更改链接，则单击检查结果中断开的链接，如图 9-7 所示，然后直接输入一个新链接即可。

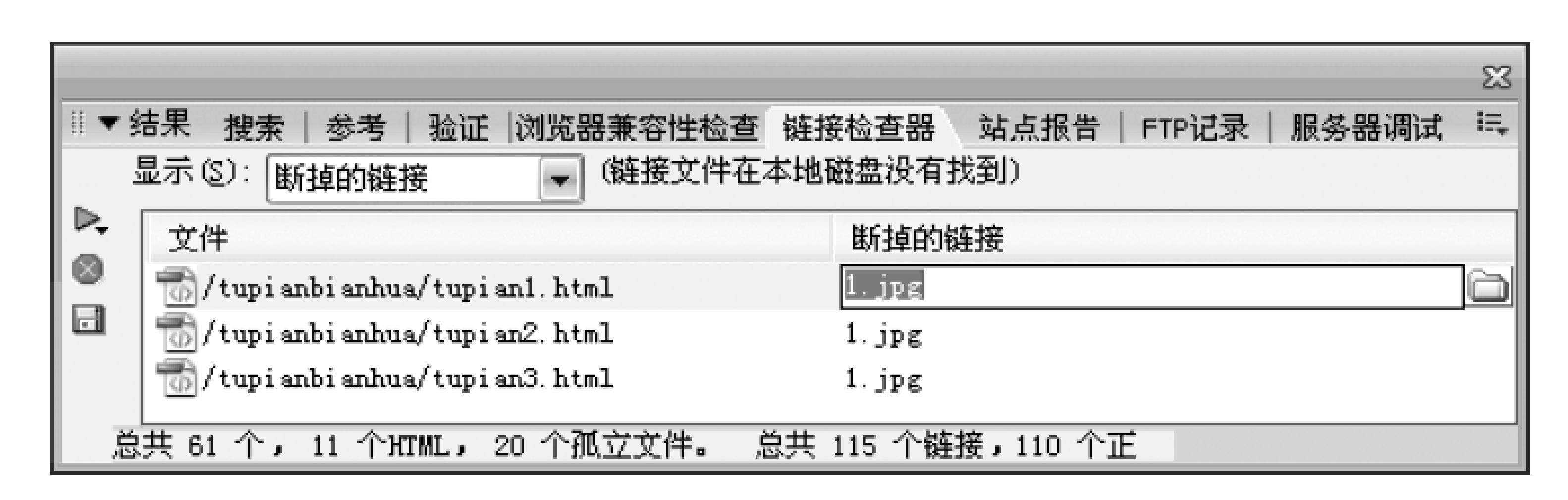


图 9-7　更改链接



（5）在【显示】下拉列表框中重新选择“孤立文件”，检查范围是“检查整个当前本地站点的链接”，则在面板中显示“孤立文件”的测试结果，如图 9-8 所示。



图 9-8 “孤立文件”测试结果

小贴士

孤立的文件只在检查整个站点时才能被查出。一般情况下，它是没用的文件（首页以及库和模板文件除外），最好删除掉，方法是：在如图 9-8 所示的“孤立的文件”列表中选中要删除的孤立文件，按 Delete 键即可。

任务 3：实施表单测试

在“新闻类别网站”中，登录页面上有表单，后台管理页面上也有表单，因此，需要进行登录表单测试和后台管理表单测试。具体操作步骤如下：

（1）登录表单测试

登录页面要求输入用户名、密码和验证码，如图 9-9 所示。网站的用户分为超级管理员、管理员和普通用户三种类型，分别拥有不同的访问权限。普通用户可以注册、登录、浏览新闻网页；管理员是网站的主要操作人员，负责网站各类数据的更新，可以进入网站后台管理页面，完成新闻数据的更新任务；超级管理员是网站权限最高的人员，负责管理网站用户权限，管理可操作网站的用户。



图 9-9 登录页面表单



进行登录表单测试时，分别以超级管理员、管理员和普通用户身份进行多次登录，测试对这三种类型的用户的身份验证是否安全，由此判断权限分割是否合理，功能实现是否正常。在输入登录信息时，既要输入正确的信息进行测试，判断能否成功登录，也要输入不正确的信息进行测试，判断能否限制非法用户的登录。

（2）后台管理表单测试

进入后台管理页面之前需要先进行登录，当管理员或超级管理员输入正确的用户名、密码和验证码后，就可以进入后台管理页面。打开每项后台功能的表单页面，进行一一测试，如图 9-10 所示。



图 9-10　后台页面表单

测试时可以按表单要求输入正确的信息和不正确的信息，查看网站功能是否含有隐藏的错误。例如，在后台添加一条新闻，然后查看主页上是否正确显示了该新闻；或者添加一个用户，测试能否以该用户信息进行登录；添加一个已存在的用户，测试系统能否容忍错误并给出错误提示；输入新用户的密码时，两次输入的密码不相同，系统能不能正确处理等。

9.1.3　相关知识拓展

在网站测试中，还有一项测试是 Cookie 测试。Cookie 是一种能够让网站服务器把少量数据储存到客户端的硬盘或内存，或是从客户端的硬盘读取数据的一种技术。

当浏览某网站时，Web 服务器会在浏览者的硬盘上存储一个非常小的文本文件，此文本文件称为 Cookie，它可以记录浏览者的用户 ID、密码、浏览过的网页、停留的时间等信息。当浏览者利用这台计算机再次来到该网站时，网站通过读取存放在硬盘上的 Cookie，得知该浏览者的相关信息，从而做出相应的动作，如在页面显示欢迎此人的提示信息，或者让此人不用再输入用户 ID、密码就直接登录等。



如果网站使用了 Cookie,就必须检查 Cookie 是否能正常工作。测试的内容可包括 Cookie 是否起作用,是否按预定的时间进行保存,网页刷新对 Cookie 有什么影响等。

9.1.4 经验总结

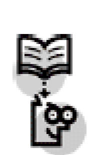
功能测试是测试中的重点,主要是测试网站的功能是否符合网站需求分析时定义的各项要求。在把站点上传服务器之前,要先在本地对其测试。实际上,在网站建设过程中,最好经常对站点进行测试并解决暴露出来的问题,要尽早发现问题并避免重犯错误。Dreamweaver 提供了多项测试站点功能,灵活使用这些功能非常有利于网站测试工作的完成。

内容测试时可以把所有文本都读一遍,这样很容易发现其中包含的错误。对于网页文字上的一些语法或拼写错误的测试,可以借助一些文字处理软件来进行,比如可以将网页上的文本暂时复制到 Microsoft Word 中进行"拼写与语法检查",或者使用一些专业的测试工具。但仅仅利用软件进行自动测试还是不够的,最后仍需要人工测试所有文本内容。

链接错误的主要原因经常是文件名不正确或路径名错误。因此,遇到链接错误时需要检查 HTML 文档中关于这个链接的引用是否正确。在 Dreamweaver 中进行过链接测试后,还需要再用手动测试全部链接。在浏览器中打开后,单击页面上的每一个链接,确定所有的链接都没有错误。

表单测试时,尤其要注意的是,测试数据一定要包含不正确的输入数据,一定要检查系统对不正确输入的反应及出错后的程序代码的处理情况。

9.2 测试网站的性能



知识储备

关于什么是一个系统的性能,举个生活实例就可以明白。现在很多地方都设有火车票代售点,虽然收点手续费,但是比跑到火车站去买票还是划算,也挺方便。可是春节前再到代售点去买票,就有问题了,售票系统很慢。这个系统平时运行没问题,而在节假日有大量购票需求的情况下,系统反应变慢的问题就是系统的性能问题。因此,如果对火车售票系统进行性能测试,就要考虑此系统最繁忙时的运行情况。一个网站本身就是一个 Web 系统,其性能是影响网站运行的重要因素,因此网站的性能测试很重要。

目前大多数网站系统都必须支持大量用户,而且网络体系架构中包含各类应用环境且由不同供应商提供软件和硬件产品,这要求系统要具有良好的性能。检测一个网站是否存在用户响应速度过慢、系统容易崩溃等问题,可以通过性能测试来实现。性能测试主要确定在用户可接受的响应时间内,系统能够承受的并发用户和并发操作的数量,以及系统在不同负载情况下页面的下载时间,检测系统瓶颈可能发生的位置,以确保系统运行的安全性、可靠性和执行效率。

网站的性能测试主要有连接速度测试和负载测试等。

(1) 连接速度测试是指测试用户打开网页的响应速度。Dreamweaver 提供了连接速度测试功能。



（2）负载测试指的是测试网站在某一负载级别上的性能，负载级别可以是在某个时刻同时访问网站的用户数量，也可以是在线数据处理的数量。例如，网站能允许多少个用户同时在线吗？如果超过了这个数量，会出现什么现象？网站能否处理大量用户对同一个页面的请求？负载测试一般采用专门的测试工具进行测试。

9.2.1　任务描述

“新闻类别网站”的测试工作任务之二：完成网站的性能测试。

9.2.2　任务实现

“新闻类别网站”的性能测试分为两个任务：

任务 1　实施连接速度测试

任务 2　实施网站负载测试

任务 1：实施连接速度测试

操作步骤如下：

（1）打开需要测试的页面，选择【编辑】|【首选参数】命令，在左侧的【分类】列表框中选择“状态栏”，在右侧的【连接速度】下拉列表框中选择 Modem 的速度，有 14.4、28.8、33.6、56、64、128 和 1500 七个参数供选择，如图 9-11 所示，此处选择 56。



图 9-11　“状态栏”参数设置

小贴士

同一个页面在不同速率的 Modem 下其下载速度是不同的，在 Dreamweaver 中可以选择不同的 Modem 对页面进行测试，了解其下载速度。如果下载速度太慢，应该考虑对页面进行修改。



（2）单击【确定】按钮回到 Dreamweaver 页面编辑状态。仔细观察状态栏，其右侧会出现一个数值“18KB/3 秒”，该数值表示当前文档大小为 18KB，大概需要 3 秒的时间可以下载完毕。重新打开图 9-11 的界面，修改 Modem 的速度设置为 128，则 Dreamweaver 状态栏的数值变为“18KB/2 秒”，表示该文档只需要大概 2 秒钟的时间就可以下载完毕。两次不同的状态栏显示连接速度如图 9-12 所示。

图 9-12　状态栏上显示连接速度

任务 2：实施网站负载测试

负载测试通常会使用自动测试工具来进行。负载测试的基本思路是：构建一个尽可能真实的运行环境，模拟大量实际用户的操作行为来产生超常的数据量和系统负载，然后监测并分析系统的各项性能指标，如 CPU 和内存的使用情况、响应时间、数据传输量等，最后提交测试报告，分析系统瓶颈，修改被测网站。

目前企业中常用的负载测试工具是 Mercury Interactive 的 LoadRunner。LoadRunner 是一个适用于各种体系架构的自动测试工具，能够模拟上千万用户实施并发负载，进行实时的性能监测，找到系统瓶颈，优化系统性能。

LoadRunner 启动后的界面如图 9-13 所示。

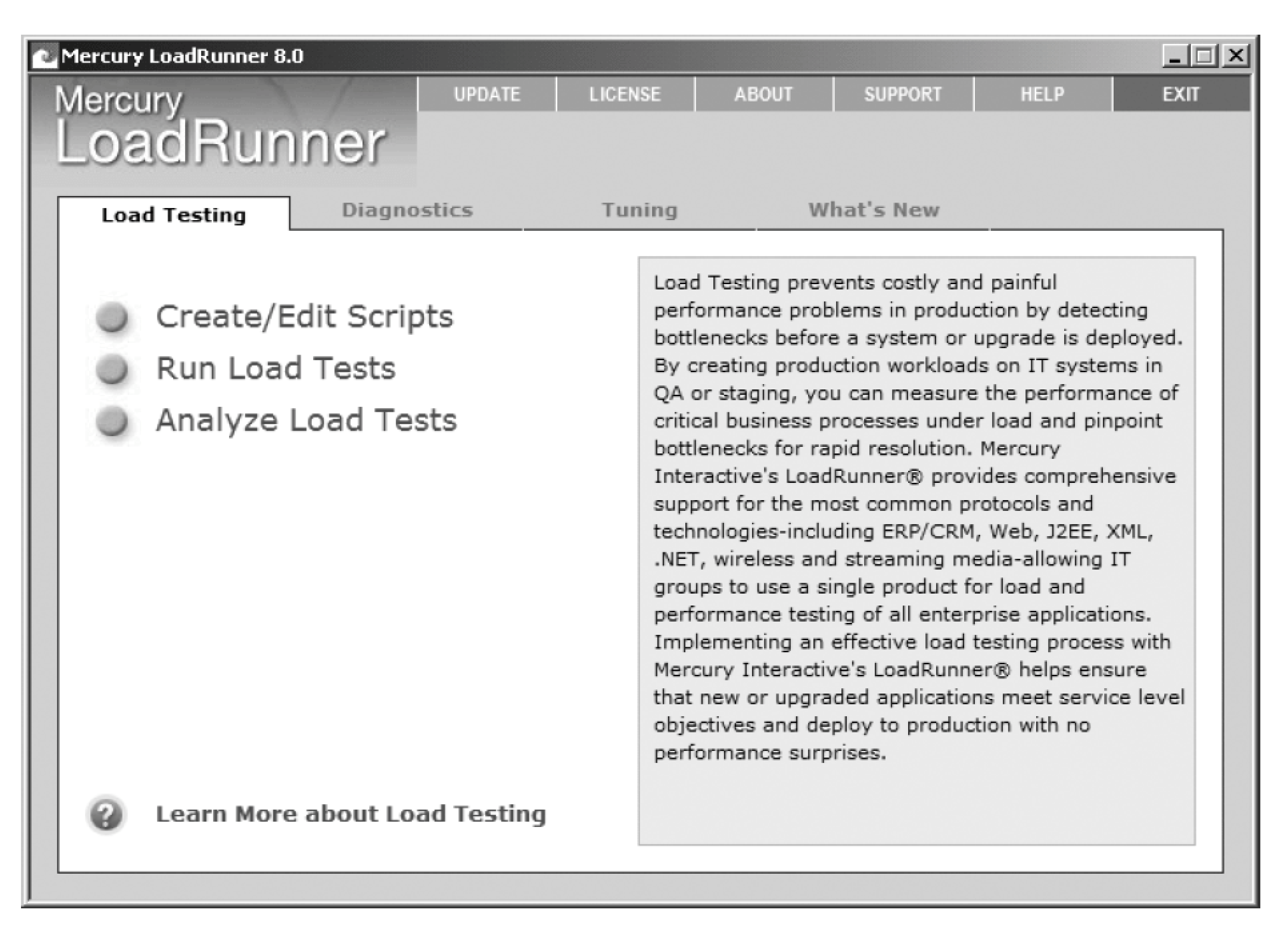


图 9-13　LoadRunner 界面

另外一个常用的测试软件是 Microsoft 的 Web Application Stress Tool(WAS)。微软提供的 WAS 虽然功能上比不了专业的 LoadRunner，但 LoadRunner 价格不菲，而微软的 WAS 则是完全免费的，而且 WAS 可以模拟 100 个强并发用户的访问，能够满足中小型网站测试的需求。



📖 小贴士

微软的 WAS 测试软件可以从网上免费下载。如果需要该软件，可以先通过搜索引擎查找 WAS 的下载链接，然后进行下载。下载后按照提示步骤安装即可。

下面以 WAS 为例，讲解使用 WAS 进行负载测试的步骤。

1. 建立测试脚本

启动 WAS 软件，程序运行时会自动打开 Create new script 对话框，即建立新脚本的窗口，如图 9-14 所示。如果运行 WAS 没有打开建立新脚本窗口，可以单击工具栏上的第一个按钮 new script 打开该窗口。

单击 Manual 按钮，建立新脚本 New Script，如图 9-15 所示。

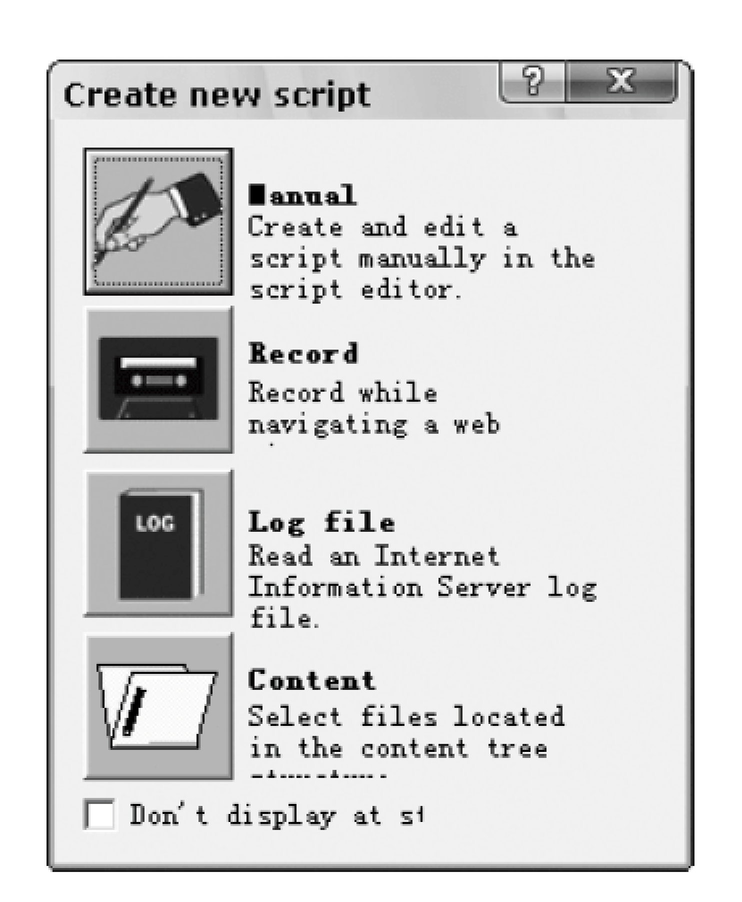


图 9-14　创建新脚本窗口

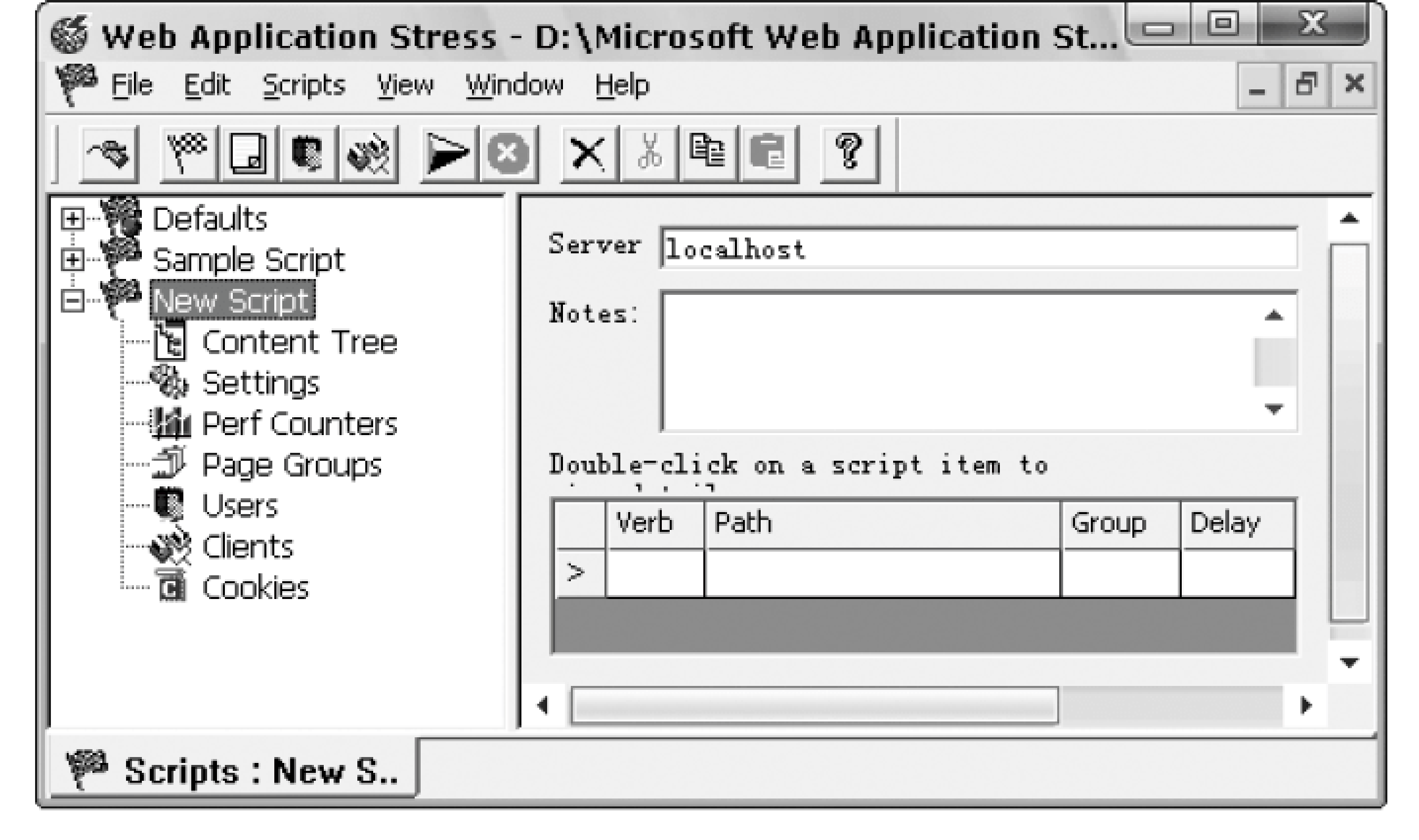


图 9-15　WAS 新脚本

初次使用时，可以在新建脚本窗口上单击 Record 按钮，打开【创建脚本向导】对话框 Browser Recorder-Step 1 of 2，如图 9-16 所示。其中，三个选项是选择要记录的内容，分别为 Request（请求）、Cookies（网上信息块）以及 Host headers（主机标题），可根据需要选择。然后单击 Next 按钮，会打开 Browser Recorder-Step 2 of 2 对话框。单击 Finish 按钮，WAS 会自动启用，并且会打开一个浏览器窗口，此时我们就可以在浏览器的地址栏中输入要测试的网站网址。随着要测试的网站内容的不断显示，在 WAS 主界面的 Recording 选项卡中的信息会实时更新。

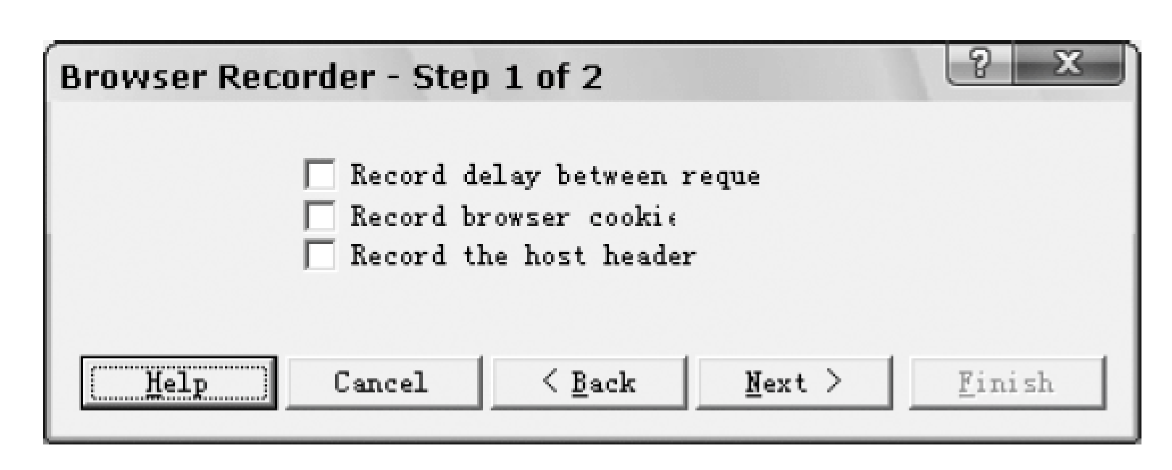


图 9-16　创建脚本向导对话框

2. 编辑脚本内容

（1）选择脚本名称，设置右侧所显示的相应选项，如图 9-17 所示。



• Server：要进行测试的服务器 IP 地址或计算机名称。
• Verb：选择脚本运行方式为 get、post 或 head。
• Path：向服务器提交的文件或字符串。

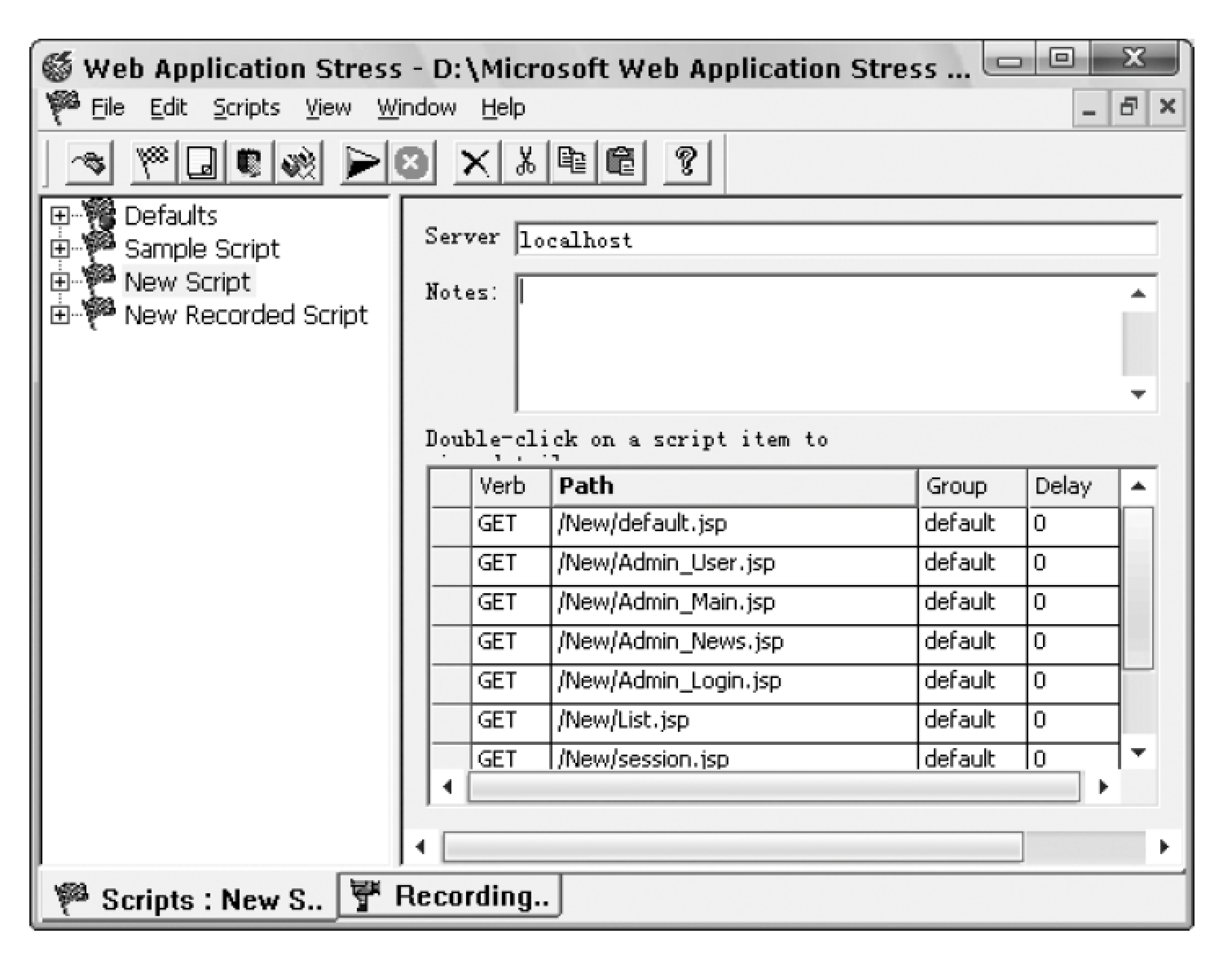


图 9-17　设置脚本选项

（2）单击脚本名称前的“＋”，展开脚本内容，选中 Settings 选项进行设置，如图 9-18 所示。

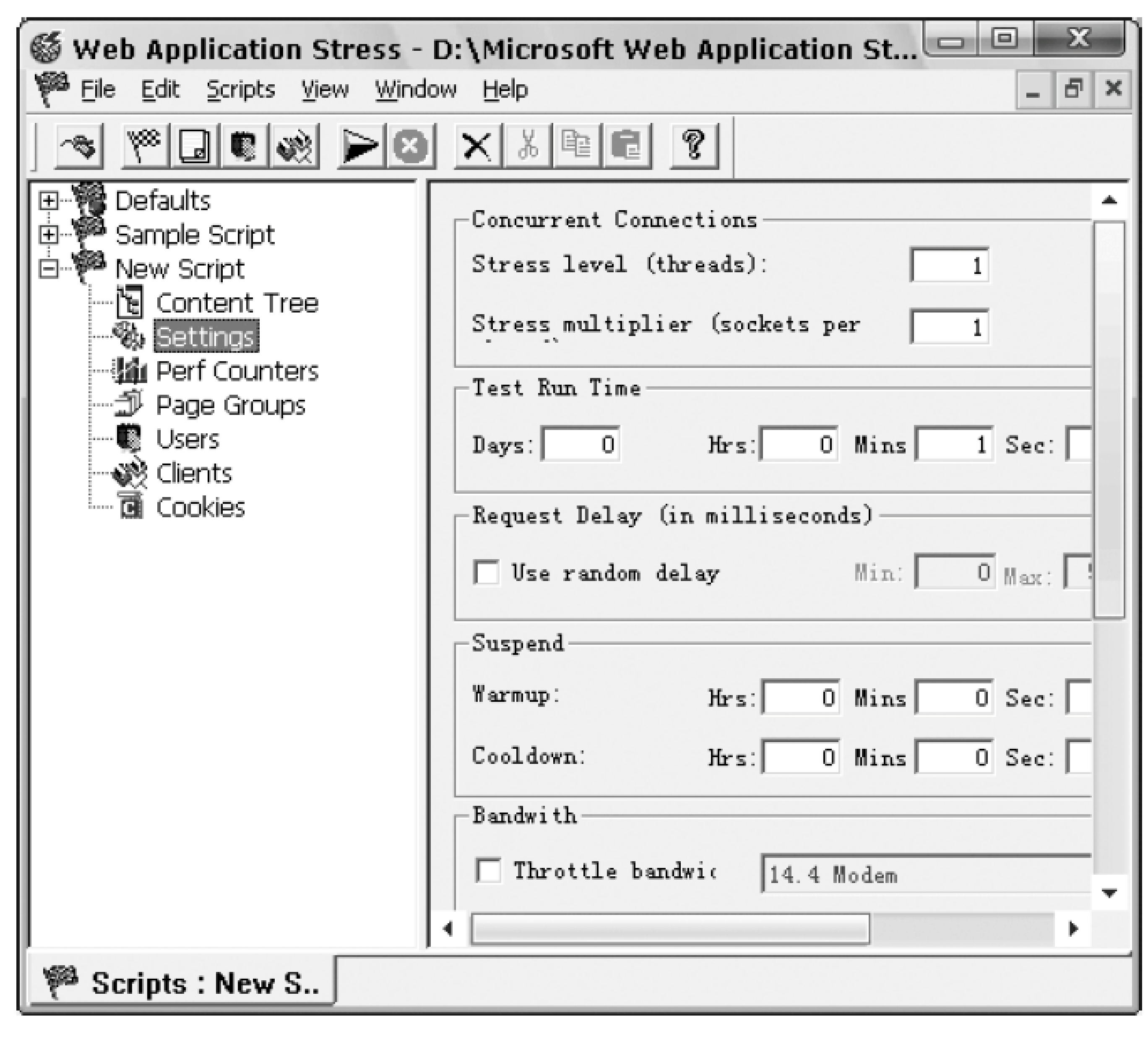


图 9-18　设置 Settings 选项

• Concurrent Connections：设置并发连接数，其下面的 Stress level (threads)选项是客户端所产生的线程数目，Stress multiplier(sockets perthread)选项是一个线程产生的 socket 请求的数目，两个数值的乘积就是客户端同时连接的并发数。



- Test Run Time：测试运行时间，以日、小时、分钟和秒来设定。建议该项时间不宜太短，如果设置的并发数较多，那么时间应该按比例增长，以便产生足够多的请求。
- Request Delay：停止时间，指连接时超出这个时间即作超时处理。
- Suspend：挂起时间，分为 Warmup 和 Cooldown 两项，一般可以设置为两三分钟，这样做的目的是避免测试开始和结束时数据的变形，影响测试的准确性。
- Bandwith：指定带宽瓶颈，即选择访问该网站大多数用户所使用的带宽。例如访问该网站的绝大部分用户是拨号，那么可以选择 56K。

(3) 在左边窗格中选中 Users 选项，双击 Default 按钮，进入用户设置页面，如图 9-19 所示。

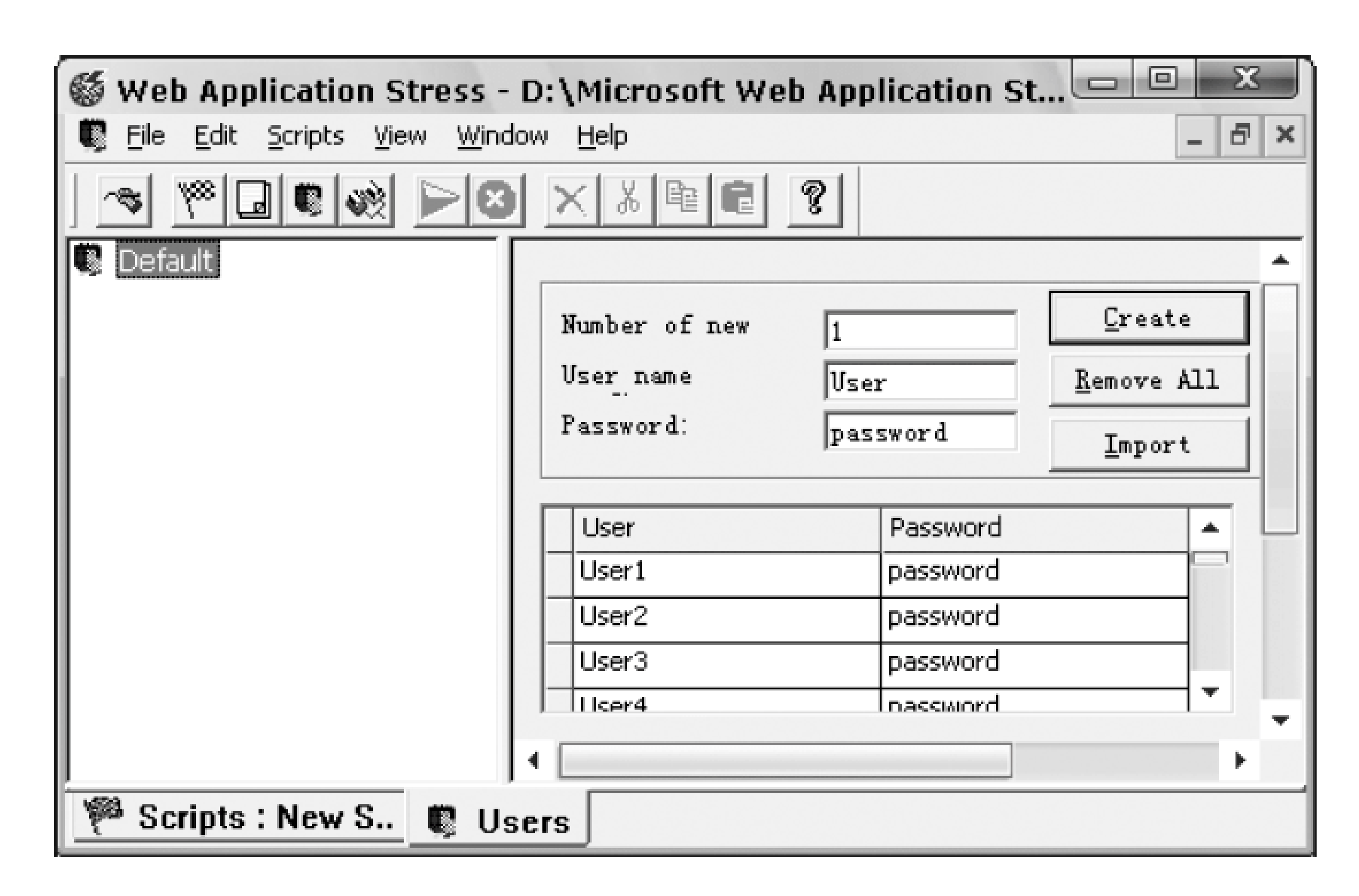


图 9-19　设置 Users 选项

在 User name 和 Password 文本框中输入服务器认可的用户和密码，单击 Create 按钮，完成添加用户功能。

(4) 在左边窗格中选中 Clients 选项，双击 Default 按钮，进入客户端设置页面，如图 9-20 所示。

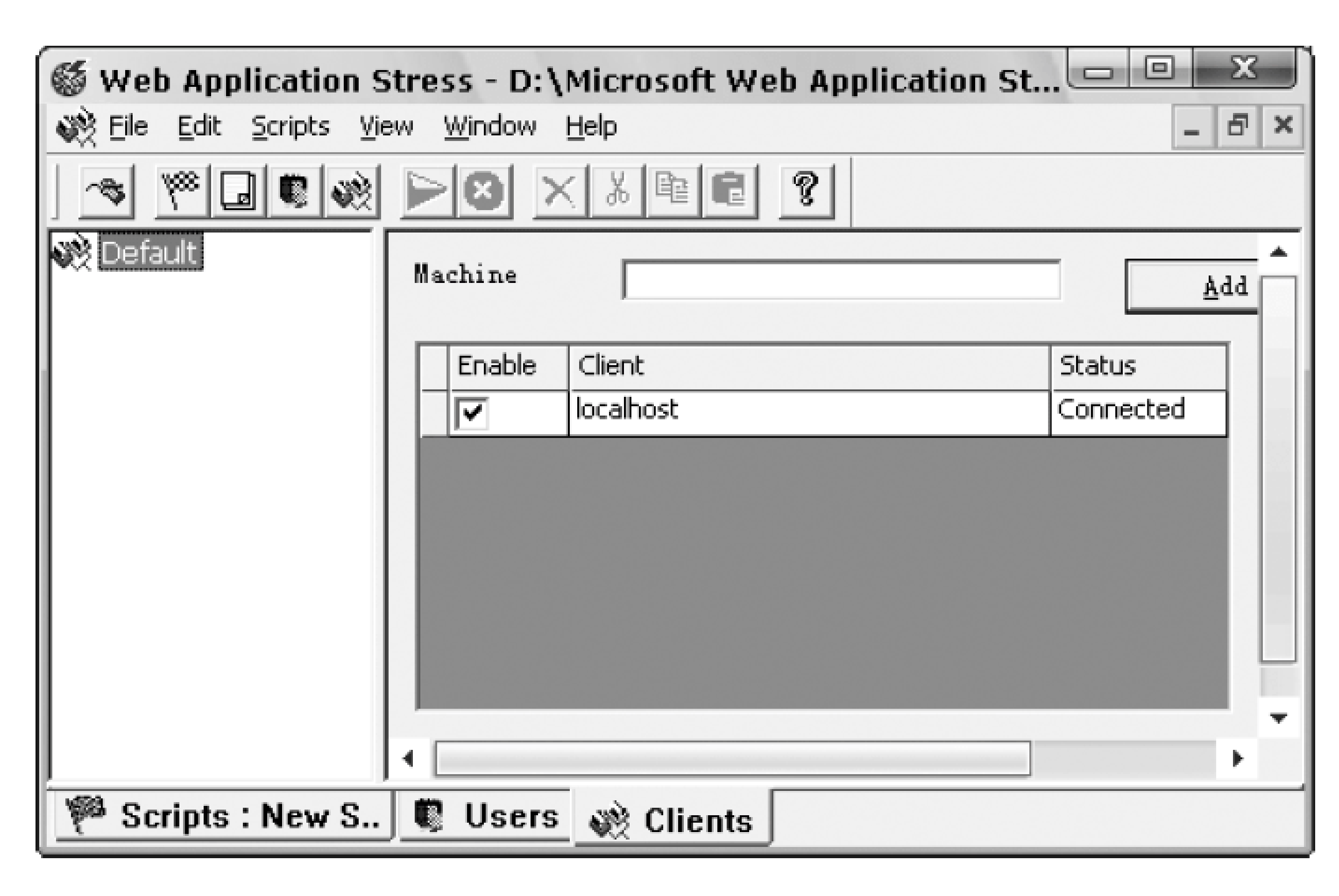


图 9-20　设置 Clients 选项



在 Machine 文本框中输入添加客户的计算机名后，单击 Add 按钮，在默认状态下只有 localhost 客户连接，添加的其他客户均离线。选中的客户会在报告中显示。

(5) 其他选项在默认情况下不用更改。

3. 运行测试

做好基本的设置工作后，在窗口左侧窗格中选中新建的脚本 New Script，单击工具栏上的 Run Script 按钮，就开始测试了。测试过程中会以进度条的方式实时显示，如图 9-21 所示。等进度条结束后，就可以进行测试结果分析了。

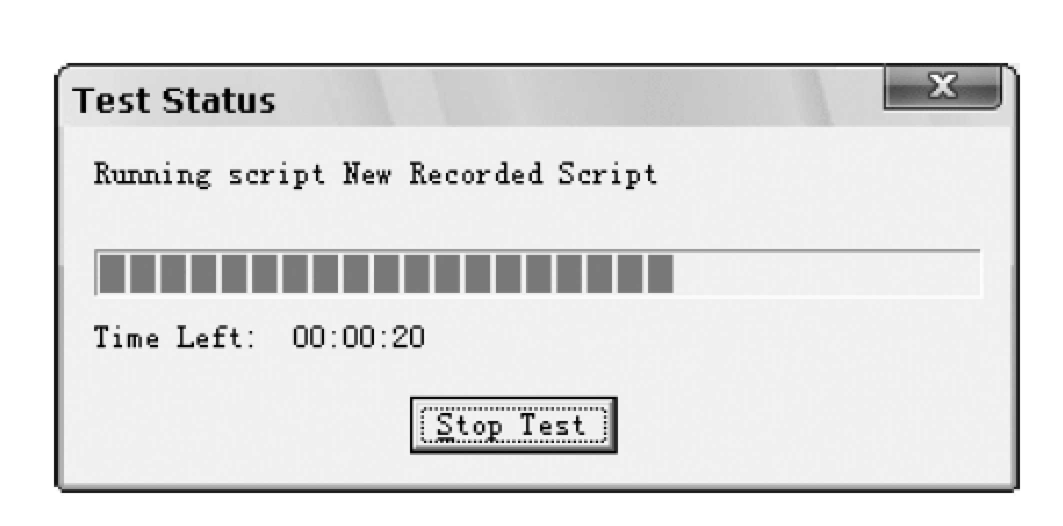


图 9-21 测试进度条

4. 测试报告

选择 View 菜单下的 Reports 命令，在打开的窗口左侧会按时间显示所有测试报告。根据时间选择本次测试报告，在窗口右侧即可查看具体内容，如图 9-22 所示。

测试报告中，重要的部分是 Socket Errors 和 Result Codes。

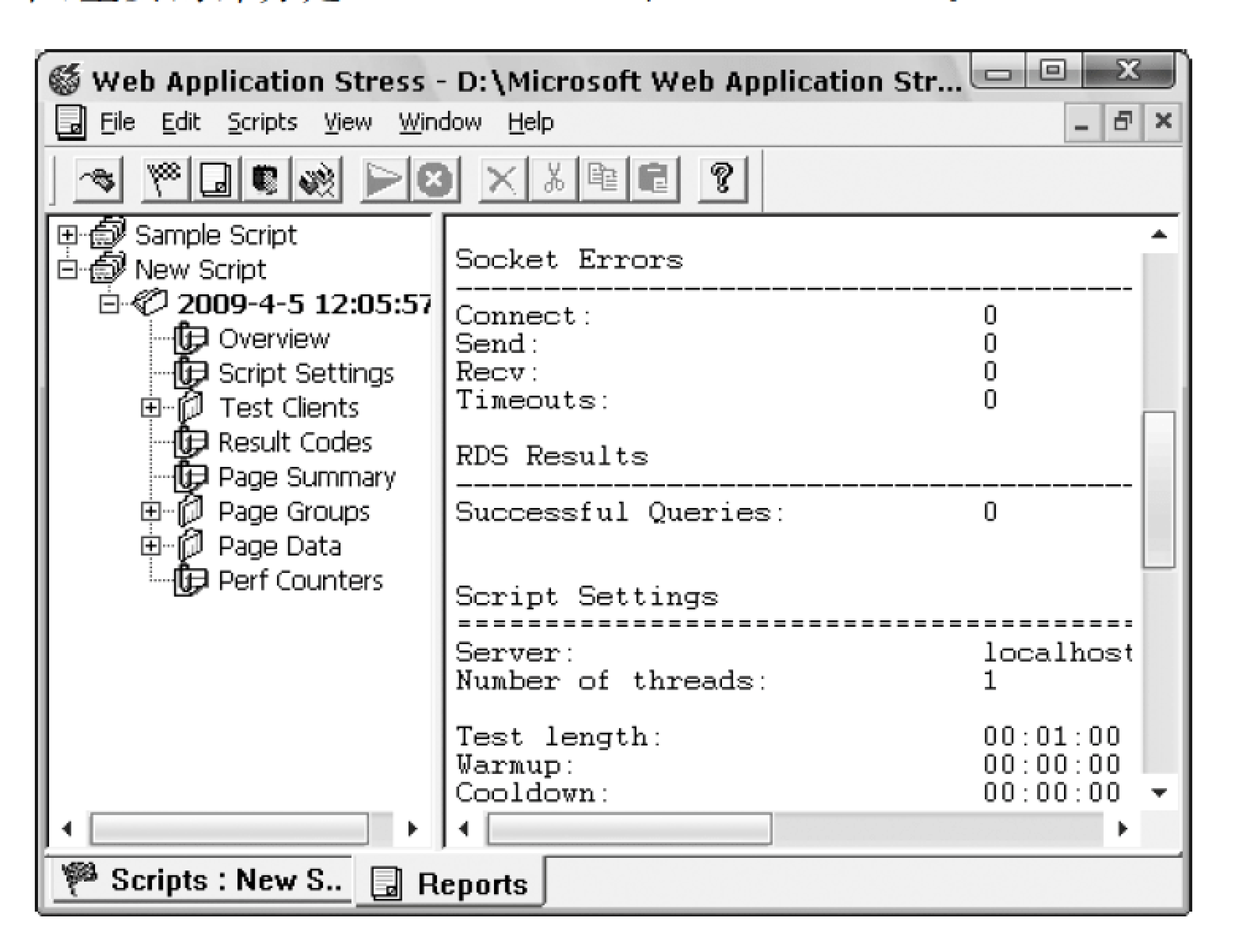


图 9-22 测试报告

Socket Errors 分为 Connect、Send 、Recv 和 Timeouts。其中，Connect 表示客户端不能与服务器取得连接的次数；Send 表示客户端不能正确发送数据到服务器的次数；Recv 表示客户端不能正确从服务器接收的次数；Timeouts 表示超时的线程数目。如果这四个数值较小或为 0，则说明服务器运行良好；如果数值较高，则说明服务器端是存在一定问题的。

在 Result Codes 中，如果 Code 列表下的数值都为 200，那么表示所有请求都经服务器成功返回；如果数值出现 400 或大于 400，例如 404，则需要在左侧窗格中找到 Page Data 节点，查看具体的错误，并进行改正。





得到测试结果后，要进行认真分析，判断网站是否存在负载问题，如果存在问题，则要找出原因所在，进一步做出修改，优化系统性能。

9.2.3　相关知识拓展

在小型网站的性能测试中，主要进行的是网站的连接速度测试。连接速度对于网站的成功与否是非常重要的，因为用户最大的抱怨经常是由于网页下载速度太慢而需要等待。不管动画多么有趣，也不管图像有多么新颖，网站速度慢是用户减少的主要因素。

那么如何提高速度呢？图像是影响速度的一个很重要因素，我们可以通过减少图像大小来优化网页速度。一个简单的规则就是，每一页的图像文件总体大小要保持在 30KB 以下。这将使网页的下载速度更快。下面是常用的减少图像大小的方法。

（1）减小物理尺寸

在图像编辑器中改变一个图像文件的物理尺寸与在 HTML 代码中改变图像的 height 和 width 属性是不同的。物理尺寸的缩减能够减少文件大小，但在 HTML 中改变图像的尺寸并不会影响文件大小。在 HTML 中改变图像的大小不仅效率非常低，而且还会使图像效果非常差。可以用图像应用程序（例如 Adobe Photoshop 或 Macromedia Fireworks）来优化文件。网页中按图像大小来指定其 height 和 width 值，这样可以让浏览器为图像预留空间。

（2）使用缩略图

如果总要下载大的图像，一种替代方法就是提供一个缩略图，如果用户想要下载大图像，单击它即可。缩略图的好处就是，小文件可以快速下载，而且如果用户想要浏览大图像也可以通过缩略图获得。

（3）图像切片

图像切片是将大图像分成多片，然后用一个表格将这些片重新组合在一起。其中，每片都可以用不同的压缩格式进行存储。有些图片可以存为 GIF 格式，有些可以存为 JPEG 格式，其压缩水平各不相同。

（4）图像文件的格式

为图像选择合适的格式也会加速网页的下载时间。照片最好存为 JPEG 格式，具有大面积纯色的图像存为 GIF 格式可以达到最优化。对上面两种图像选择 PNG 格式也不错，但是老版本的浏览器不支持 PNG 格式。

9.2.4　经验总结

性能测试对于网站的运行而言非常重要。进行性能测试时，假设的客户端一定要比实际可能的数量还要大，这样才能测试出真正的效果。通过性能测试，了解网站存在的性能问题后，应该想办法提高网站性能。

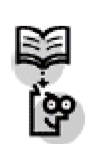
对网站做连接速度测试时，除了可以利用 Dreamweaver 自带的测试工具外，还可以使用实际环境运行网站的方式来进行测试。在实际环境中测试网站各页面的实际下载时间，从浏览者的角度来了解页面的下载速度，特别是测试类似主页之类的重要页面。如果有条件，还应该使用拨号、宽带等各种上网方式测试网页下载的情况。



连接速度测试完成后，根据测试结果的值，如果需要调整网页，可以考虑页面文件的大小、插入图片的分辨率、图像切片大小和脚本语言等影响下载时间的因素，最终目的是减少下载时间，让用户在最短的时间内看到页面。

为了完整反映一个网站在服务器上的运行情况，负载测试需要进行不同的设置，并且要进行多次测试，以保证网站在需求范围内能正常运行。

9.3 其他专项测试



知识储备

其他专项测试包括兼容性测试、可用性测试、安全性测试等。

1. 兼容性测试

兼容性测试是指测试某个软件在特定的硬件平台上、不同的操作系统平台上、不同的应用软件之间、不同的网络环境之间能否正常运行。兼容性测试无法做到完全的质量保证，但可以保证网站能满足大多数用户的要求。

网站兼容性测试主要是针对不同的操作系统平台、不同的分辨率、不同的浏览器进行测试。由于无法确定用户在客户端会使用什么样的主机访问网站，因此，进行网站兼容性测试就是必不可少的一个步骤。

(1) 操作系统兼容性测试

市场上有很多不同的操作系统类型，常见的有 Windows、UNIX、Macintosh、Linux 等。非服务器计算机上最常用的是 Windows 操作系统。Windows 操作系统包括 Windows XP、Windows 2003、Vista、Windows 2000/NT，Windows 9X 等。进行操作系统兼容性测试的主要目的就是尽可能保证网站在各种操作系统平台上都能正常运行。

(2) 分辨率兼容性测试

屏幕分辨率是指屏幕横向纵向所能显示的像素数量。不同分辨率下的网页显示是不一样的。例如，将屏幕设成 800×600 像素的分辨率，和设成 1024×768 像素相比，前者使屏幕显示的像素点少了，这样字就会变大，图像也会变大。

分辨率兼容性测试是为了使页面版式在不同的分辨率模式下能正常显示、字体能符合网页设置的要求而进行的测试。网站浏览者使用什么模式的分辨率是未知的，因此需要对经常用到的分辨率分别进行测试。

(3) 浏览器兼容性测试

浏览器是 Web 系统中最核心的构件，来自不同厂家的浏览器对 JavaScript、ActiveX 或不同的 HTML 规格有不同的支持，即使是同一厂家的浏览器，也存在不同的版本问题。不同的浏览器对安全性和 Java 的设置也不一样。

浏览器兼容性测试是指在不同的浏览器和不同浏览器版本下，测试页面的运行和显示情况，要确保网站在常用的浏览器中能够正常显示。不同浏览器测试在 Dreamweaver 中非常简单，它能将测试出的错误或可能出现错误的地方列出一个报告单，然后根据该报告单的提示进行修改和处理，解决有可能出现的问题。

2. 可用性测试

可用性测试，简单说就是测试网站是否易于使用。为了测试网站上的各种要素是否满



足网站各级用户的使用要求,可以选择一些有代表性的网站用户在实际环境中对网站进行访问和操作,同时测试人员在一旁观察、聆听、做记录,以了解用户对网站的反应,发现一些网站设计上的问题。

可用性测试包括页面外观测试、导航信息测试和图形图像测试等,以手动测试为主。

(1) 页面外观测试

页面外观测试就是通过浏览器浏览每一个网页,检查网站的总体设计是否合理,是否一切都很和谐,是否能够达到传达信息和主题思想的目的,是否能够满足浏览者的需要。

(2) 导航信息测试

导航是否直观、网站主要内容是否可通过主页存取、网站是否需要站点地图、网站是否需要搜索引擎或其他的导航帮助等,是决定一个网站是否易于导航的主要因素。导航的另一个重要方面是网站的页面结构、导航、菜单、链接的风格是否一致。实施导航信息测试时需要了解浏览者是否能凭直觉知道网站里有没有可取的内容,内容在什么地方。

(3) 图形图像测试

在一个网站中,适当的图片和动画既能起到广告宣传的作用,又能起到美化页面的功能。一个网站的图形可以包括图片、动画、边框、颜色、字体、背景、按钮等。常见的图形图像测试包括:

① 要确定图形有明确的用途,图片或动画不要胡乱地堆在一起,以免浪费传输时间。图片尺寸要尽量地小,并且要能清楚地说明某件事情,一般都链接到某个具体的页面。

② 验证所有页面字体的风格是否一致。

③ 确定背景颜色是否与字体颜色和前景颜色相搭配。

④ 验证图片的大小和质量是否符合要求,一般采用 JPG 或 GIF 压缩。

3. 安全性测试

随着网络应用的越来越普及,网站安全性越来越重要。安全性测试是用来查找网站程序设计中是否存在安全隐患的。

(1) 注册与登录。现在很多网站都采用先注册、后登录的方式。因此,要测试有效和无效的用户名和密码,注意是否存在大小写敏感,限制可以尝试多少次登录,是否可以不登录而直接浏览某个页面等。

(2) 在线超时。测试网站浏览是否有超时的限制,也就是说,用户登录一定时间内(例如 15 分钟)没有点击任何页面,是否需要重新登录才能正常使用。

(3) 日志文件。为了保证 Web 应用系统的安全性,日志文件是至关重要的。需要测试相关信息是否写入了日志文件,是否可以追踪。

(4) 备份与恢复。为了防范系统意外崩溃造成的数据丢失,备份与恢复应该是一个必备功能。需要测试所采用的备份与恢复方式是否有效。

(5) 信息加密。当使用了安全套接字时,还要测试加密是否正确,检查信息的完整性。

(6) 服务器端的脚本漏洞检查。存在于服务器端的脚本常常构成安全漏洞,这些漏洞又往往被黑客利用。所以,还要测试没有经过授权,就不能在服务器端放置和编辑脚本的权限问题。

9.3.1 任务描述

"新闻类别网站"的测试工作任务之三:完成网站的其他专项测试。



9.3.2 任务实现

“新闻类别网站”的其他专项测试分为三个任务：

任务 1　兼容性测试

任务 2　可用性测试

任务 3　安全性测试

任务 1：兼容性测试

1. 实施操作系统兼容性测试

在浏览者常用的操作系统上运行“新闻类别网站”，对每个页面每个表单功能都要进行测试，测试重点是：检查网站的运行在当前的操作系统上是否会出现兼容性异常。具体操作步骤如下：

（1）在 Windows XP 平台上运行“新闻类别网站”，修正所有的操作系统兼容性异常。

（2）在 Linux 平台上运行“新闻类别网站”，修正所有的操作系统兼容性异常。

小贴士

网站浏览者究竟使用哪一种操作系统，取决于其所使用的计算机系统的配置。网站浏览者使用操作系统的种类数，直接决定了网站操作系统兼容性测试时需要考虑的操作系统数量。执行操作系统兼容性测试时要分别在安装了不同类型操作系统或不同版本操作系统的计算机上运行网站，查看是否存在兼容性问题。当然，也可以用一些兼容性测试工具进行不同操作系统的测试。

2. 实施分辨率兼容性测试

目前常见的显示器分辨率有三种：800×600、1024×768 和 1280×1024。在通常情况下，在网站的需求规格说明书中会建议使用某些分辨率。对于测试来讲，要针对这些分辨率进行专门的测试。分辨率兼容性测试比较容易，可以通过设置屏幕分辨率属性来进行不同的分辨率测试。具体操作步骤如下：

（1）设置屏幕分辨率为 800×600，在其上运行“新闻类别网站”，查看网页是否能够按照网站需求分析的要求正常显示，并修正所有的分辨率兼容性异常。

（2）设置屏幕分辨率为 1024×768，在其上运行“新闻类别网站”，查看网页是否能够按照网站需求分析的要求正常显示，并修正所有的分辨率兼容性异常。

（3）设置屏幕分辨率为 1280×1024，在其上运行“新闻类别网站”，查看网页是否能够按照网站需求分析的要求正常显示，并修正所有的分辨率兼容性异常。

小贴士

设置屏幕分辨率：在 Windows XP 操作系统中，可以在桌面空白处单击鼠标右键，然后从弹出的快捷菜单中选择【属性】命令，打开【显示属性】设置对话框。单击【设置】选项卡，然后在【屏幕分辨率】及【高级】中进行调整，如图 9-23 所示。在 Windows Vista 操作系统中，同样在桌面空白处单击鼠标右键，再选择【个性化】命令，然后单击【个性化】界面中的【显示属性】选项，具体设置与 Windows XP 一样。



图 9-23　设置分辨率

3. 实施浏览器兼容性测试

利用 Dreamweaver 中自带的测试工具完成“新闻类别网站”的浏览器兼容性测试。具体操作步骤如下：

(1) 单击【结果】面板中的【浏览器兼容性检查】选项卡，转换到【浏览器兼容性检查】选项卡中。单击【结果】面板左侧的绿色三角箭头按钮，弹出下拉菜单，如图 9-24 所示。

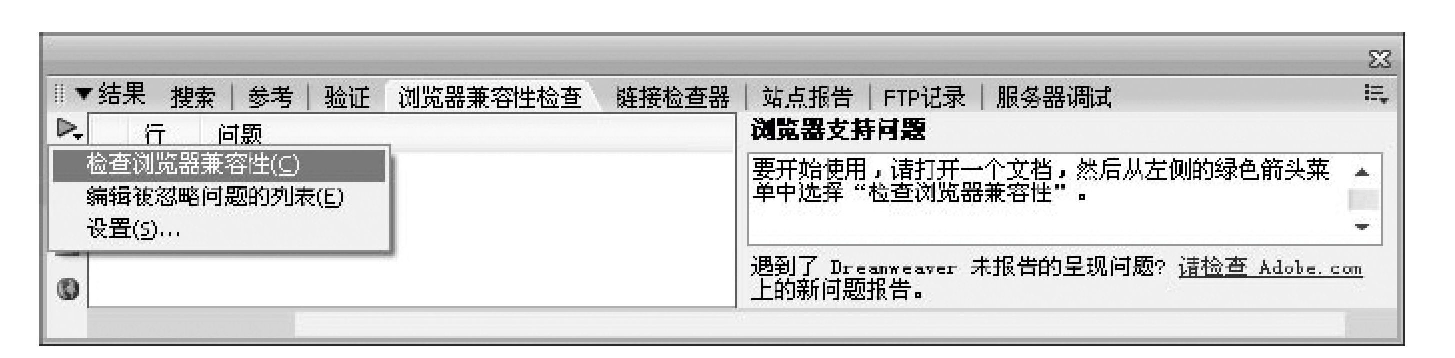


图 9-24　【浏览器兼容性检查】选项卡

(2) 在弹出的下拉菜单中选择【设置】命令，弹出【目标浏览器】对话框，可用来选择不同的浏览器和版本，如图 9-25 所示。此处的选择原则是选择版本较低的浏览器进行测试，因为新版本浏览器一般都支持旧版本的浏览器，选择后单击【确定】按钮返回。

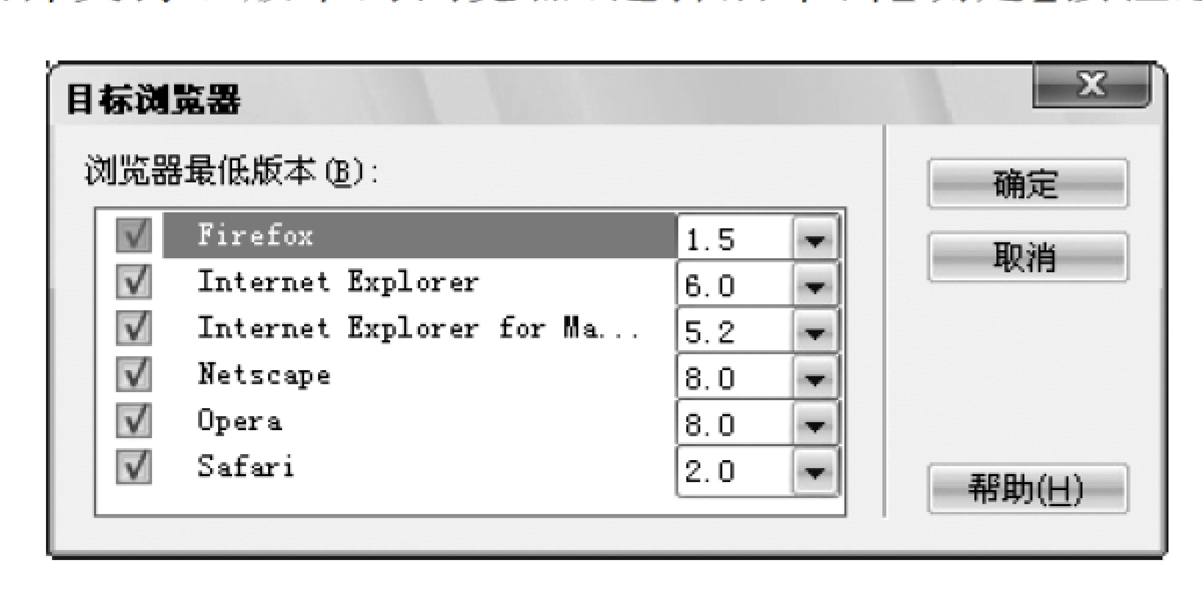


图 9-25　【目标浏览器】对话框



（3）选择图 9-24 中的【检查浏览器兼容性】命令，Dreamweaver 会自动对当前本地站点中的所有文件进行目标浏览器的检查，并显示出检查结果，如图 9-26 所示。报告单中列出了错误项、警告项和有可能出现的错误项，并在报告的后面列出了出现错误的具体位置和原因。



图 9-26　站点报告

小贴士

在进行目标浏览器检查时，既可以对整个站点进行检查，也可以设置检查某一个页面或某些文件。

现在，很多网站的运行都依赖两种技术——Flash 和 JavaScript。但并不是所有人都会在浏览器上启用这两项功能，从而导致网站的某些 Flash 和 JavaScript 内容不能正常显示，因此，网站还应该在浏览器端打开和关闭 Flash 与 JavaScript 的情形下进行浏览器兼容性测试。

任务 2：可用性测试

1．实施页面外观测试

具体操作步骤如下：

（1）验证“新闻类别网站”各个页面的风格是否一致，是否美观大方。

（2）验证“新闻类别网站”页面显示内容是否完整，当测试数据的记录数超过一屏/一页时，验证满屏/页时其窗体是否有横向、纵向滚动条或换页打印，界面显示是否正常。

（3）验证“新闻类别网站”页面是否友好，数据是否是按网站浏览者的习惯进行分类、排序，进行某些操作时是否有提示信息，各种操作是否方便易用。

2．实施导航信息测试

具体操作步骤如下：

（1）检查“新闻类别网站”的导航是否直观，网站的主要内容是否可通过主页存取。该网站是否有站点地图，是否有搜索引擎。根据检查结果，修正不符合要求的网页。

（2）检查“新闻类别网站”的页面结构、导航、菜单、链接的风格是否一致。根据检查结果，修正不符合要求的网页。

（3）寻找有代表的“新闻类别网站”的浏览者，了解他们是否能凭直觉知道网站里有没有可取的内容，内容在什么地方。根据检查结果，修正不符合要求的网页。

3．实施图形图像测试

具体操作步骤如下：

（1）检查“新闻类别网站”的图形图像，确定所用的图形图像都有特定的用途，所指向的



链接都能正常显示。根据检查结果，修正不符合要求的图形图像。

(2) 检查“新闻类别网站”的各个页面、字体风格是否一致。根据检查结果，修正不符合要求的页面字体。

(3) 检查“新闻类别网站”的背景色、前景色、字体颜色是否协调。根据检查结果，修正不符合要求的页面。

(4) 检查“新闻类别网站”中所使用的图片大小与图片质量是否符合网站需求分析的要求。根据检查结果，修正不符合要求的图片。

任务 3：安全性测试

(1) 对“新闻类别网站”的注册与登录界面进行测试，检查对注册时的无效用户名系统能否识别并处理，检查登录时有没有尝试登录的次数限制，检查是不是存在不登录就可以访问的页面。对不符合要求的网页进行修正。

(2) 检查“新闻类别网站”服务器端是不是存在脚本漏洞，是否能够限制未授权的人不能在服务器上放置和编辑网站脚本。如果不符合要求则进行修正。

9.3.3　相关知识拓展

1. 测试报告

测试工作结束时要编写测试报告。

测试报告是把测试的过程和结果写成文档，并对发现的问题和缺陷进行分析，为纠正软件存在的质量问题提供依据，同时为软件验收和交付打下基础。

测试报告中应包含的内容：

(1) 测试标题。标题通常是×××网站的×××测试报告。

(2) 测试目的。说明进行的网站测试的主要目的，测试系统是否符合需求(或达到×××功能目标)，发现网站可能存在的一些问题。

(3) 测试项目简介。包括项目需求、系统框架图、网络拓扑图等。

(4) 测试方法和工具。简要介绍测试中采用的方法，写上测试重点和测试模式。如果采用了测试工具，则要说明测试工具是自产还是其他厂商生产的，版本号是多少。

(5) 测试环境与配置。简要介绍数据库服务器配置、应用服务器配置、客户端配置、网络环境等，对于三层架构的系统，可以根据网络拓扑图列出相关配置。

(6) 测试机构和人员。列出测试机构和主要测试人员。

(7) 测试执行过程。描述测试执行的全过程，按顺序给出每一项测试的具体步骤、结果数据、表明的事实以及发现的问题。

(8) 测试结果和建议。说明测试目标是否完成、测试是否通过、总结系统中存在的问题，针对问题提出改进建议。

2. Linux 简介

Linux 是一种计算机操作系统，中文读法一般为“哩纽克斯”，但真正的读法应为“哩纳克斯”。Linux 是一套免费使用和自由传播的类 UNIX 操作系统，它主要用于基于 Intel X86



系列 CPU 的计算机上。

Linux 的出现，最早开始于一位名叫 Linus Torvalds 的计算机业余爱好者，当时他是芬兰赫尔辛基大学的学生。他的目的是想设计一个代替 Minix（是由一位名叫 Andrew Tannebaum 的计算机教授编写的一个操作系统示教程序）的操作系统，这个操作系统可用于 386、486 或奔腾处理器的个人计算机（Personal Computer，PC）上，并且具有 UNIX 操作系统的全部功能，因而开始了 Linux 雏形的设计。

Linux 的基本思想有两点：第一，一切都是文件；第二，每个软件都有确定的用途，同时它们都尽可能被编写得更好。其中第一条详细来讲就是系统中的所有都归结为一个文件，包括命令、硬件和软件设备、操作系统、进程等，它们对于操作系统内核而言，都被视为拥有各自特性或类型的文件。至于说 Linux 是基于 UNIX 的，很大程度上也是因为这两者的基本思想十分相近。



图 9-27　Linux 的企鹅标志

Linux 的标志和吉祥物是一只名字叫做 Tux 的企鹅，标志的由来是因为 Linux 在澳洲时曾被一只动物园里的企鹅咬了一口，便选择了企鹅作为 Linux 的标志，如图 9-27 所示。

Linux 以它的高效性和灵活性著称。它能够在 PC 计算机上实现全部的 UNIX 特性，具有多任务、多用户的能力。Linux 操作系统软件包不仅包括完整的 Linux 操作系统，还包括了文本编辑器、高级语言编译器等应用软件。它还包括带有多个窗口管理器的 X-Window 图形用户界面，如同使用 Windows NT 一样，允许使用窗口、图标和菜单对系统进行操作。

Linux 之所以受到广大计算机爱好者的喜爱，主要原因有两个，一是它属于自由软件，用户不用支付任何费用就可以获得它和它的源代码，并且可以根据自己的需要对它进行必要的修改，无偿使用，无约束地继续传播。另一个原因是，它具有 UNIX 的全部功能，任何使用 UNIX 操作系统或想要学习 UNIX 操作系统的人都可以从 Linux 中获益。

目前，中国内地的 Linux 发行版有：

红旗 Linux（Redflag Linux）、冲浪 Linux（Xteam Linux）、蓝点 Linux、Hiweed GNU/Linux、Magic Linux、Engineering Computing GNU/Linux、Open Desktop、新华 Linux、共创 Linux、百资 Linux、中标普华 Linux、中软 Linux、新华华镭 Linux（RaysLX）、CD Linux、MC Linux、即时 Linux（Thizlinux）、b2d linux、IBOX、MCLOS、FANX、Everest、酷博 linux。

9.3.4　经验总结

在做测试时，要根据网站的特点选择合适的测试过程，这样才能收到好的效果。网站测试的目的就是发现网站中存在的问题。事实上，用户对于网站不满意的很多问题都是很容易发现并解决的，但是不通过测试可能就发现不了这些问题，解决这些问题更是无从谈起。

这正是测试的重要性所在。

一般情况下，至少需要在 Windows 平台和 Linux 平台上进行兼容性测试。对于操作系统的不同版本，应当针对当前主流操作系统版本进行兼容性测试，在确保主流操作系统版本兼容性测试的前提下，再对非主流操作系统版本进行测试，尽量保证操作系统版本兼容性测试的完整性。

导航信息的功能是引导浏览者在页面上搜索查找信息，并实现各个页面之间的跳转。测试时要看导航是否直观，是否易于导航。对于大多数浏览者而言，是没有心思和精力去搞清楚某个网站的整体结构的，一般在浏览时凭的是第一眼的感觉，如果无法快速找到信息，就会很快离开。因此，网页上的导航信息要尽可能地简单准确。

关于参加可用性测试的人员，不应该选择网页设计人员，最好是找一些与网站开发没有关系的人员或网站的各类用户进行测试。因为网页设计人员很容易认为所有的设计都是完美的，但事实并非如此。测试时要让人们不带任何偏见地浏览网站，提出宝贵意见。利用可用性测试的反馈信息，可以更好地改进网站的设计。

下面给出了一些测试问题，可以在测试时使用。

- 你对网站的第一印象如何？
- 你认为网站的目的是什么？
- 进入网站后，你想去哪儿？
- 你是否能够轻松确定如何去查找某一主题的更多信息？
- 网站组织结构是否合理？
- 网站的所有元素（例如动画、视频、表单）的运作是否与你的期望一致？
- 对于这个网站，你喜欢什么？不喜欢什么？
- 查找特定主题内容时，你是否会遇到麻烦？
- 你所处的位置在哪里？你是如何判断的？
- 怎样返回首页（主页）？

9.4　实　　训

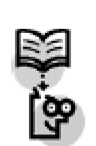
对“新闻类别网站”进行功能测试、性能测试和其他专项测试。



第10章　发布网站

在本地的计算机上制作好网站，并通过测试之后，还需要将其发布到 Internet 上，才能被 Internet 的浏览者所访问。在网站发布之前，需要先在 Internet 上申请域名和空间，然后利用 FTP 软件上传网站。站点发布之后，还需要对网站进行宣传，提高网站的访问率。

10.1　申请域名和空间



知识储备

1. IP 地址与域名

Internet 上的每一台主机都有一个 IP 地址。IP 地址是一个 32 位的二进制数，为方便使用，用四个点分十进制数来表示(如 202.108.22.46)。但即使如此，直接使用或者记忆 IP 地址仍很不方便，因此，产生了使用比较直观的文字符号或数字来代替 IP 地址的办法，这就是域名。域名是用来替代 IP 地址以方便浏览者访问的，相对于 IP 地址而言，域名更便于理解和记忆。例如，浏览者想访问百度网站，只需输入 www.baidu.com 就可以了，www.baidu.com 就是域名。

2. 域名与网站类型

域名是由多个字段组成(一般为三个字段或四个字段)，各字段之间用英文的句号“.”来分隔，如 www.sina.com.cn，这个域名分为四个字段。第一段表示入网的主机名，一般命名为 www，表示这台主机提供 www(万维网)服务；第二段表示这台主机所处的网络的名称，比如 sina；第三段 com 表示是 sina 这个网络是商业类型的网络，又称为二级域名；第四段表示二级域名所属的国家，cn 表示 sina 所处的这个商业网络属于中国，第四段又称为顶级域名。

对于三段式域名，比如 www.baidu.com，www 表示主机名，baidu 表示此主机所处网络的名称，com 表示此网络属于国际商业网络。

所以，网站域名的倒数第二段或最后一段表示了网站的类型。

3. 域名注册与解析

域名是由国际域名管理组织或国内的相关机构统一管理的。有很多网络公司可以代理域名的注册业务，例如中国万网 http://www.net.cn、中国频道：http://www.chinachannel.com、时代互联 www.now.cn、新网 http://www.xinnet.com 等，可以在搜索引擎上查找到域名代理商，在它们的公司网站上注册域名。

域名需要指向网站服务器的 IP 地址。在注册域名之后，需要把域名解析到一个指定的 IP 上。一个域名可以解析多个二级域名，这些二级域名可以指向不同服务器的网站。除了解析主机名之外，域名还可以作为别名、邮件交换记录的解析。

4. 服务器空间

服务器空间就是存放网站文件的地方。服务器空间可分为两大类型，免费空间和付费空间。由于免费空间的服务不能得到保证，企业一般都会申请付费空间。企业经常采用的服务器形式有三种：虚拟主机、服务器托管和自建服务器。

(1) 虚拟主机是使用特殊的软硬件技术，把一台网站服务器划分为若干个“虚拟”的主机。每个虚拟主机都可以是一个独立的网站，可以具有独立的域名，具有完整的 Internet 服务器功能（WWW、FTP、E-mail 等）。同一台主机上的虚拟主机之间是完全独立的，在外界看来，每一台虚拟主机和一台独立主机的表现完全一样。

(2) 服务器托管是指在专门的服务器托管公司环境里放置一台服务器，或向其租用一台服务器，满足客户远程控制、WWW 浏览、E-mail 收发、FTP 文件的上传与下载等功能。服务器托管的基本形态就是租用专门的服务器托管公司的网络线路、自己选择服务器配置及服务应用软件，然后由专门的服务器托管公司负责维护。

(3) 自建服务器是指企业自己准备硬件资源，自己安装服务软件，并自行维护。

10.1.1　任务描述

“新闻类别网站”发布任务之一：申请域名和服务器空间。

小贴士

一般来讲，网站的发布分为两种情况，其一是所开发的网站作为已有站点的一个版块；其二是所开发的网站作为一个独立的新网站。前者的发布比较简单，只需要将网站代码上传至网络管理员指定的服务器位置，并在已有的网站中建立一个指向其首页的链接即可。对于后者稍麻烦些，本节内容以后者为例介绍网站的发布操作。

服务器空间用于存放站点文件夹，域名用于访问网站。

10.1.2　任务实现

申请域名和服务器空间包括两个任务：

任务 1　申请网站域名

任务 2　申请服务器空间

任务 1：申请网站域名

(1) 确定域名类型

根据“新闻类别网站”的特点，确定网站类型，即确定其域名的最后一段或最后两段（又称为域名的后缀）。

注意

这里只给出工作步骤，不给出工作结果，请读者自行确定域名后缀。



（2）选取网站域名

为“新闻类别网站”选取合适的域名。

小贴士

域名具有全球唯一性。在选取了某个域名之后，必须确定该域名是否已经被人使用。可以到提供域名注册的站点上进行域名查询。如果发现该域名已经存在，就必须选一个新的域名。域名注册网站都提供域名查询功能，可以对要申请的域名进行查询。

（3）注册域名

为“新闻类别网站”注册域名。

注意

这里只给出工作步骤，不给出工作结果，请读者自行注册网站域名。

在选定一个尚未使用的域名作为自己的域名之后，就可以注册了。

域名注册时，注册与交费都在网上完成，一般不需要在网络公司实际办理。网上域名注册的步骤大致相同，一般包括域名查询、类型选择、年限选择、信息填写、信息确认、缴纳费用等，每一步骤的操作在相关网页上都有提示。

下面以在中国万网上注册一个域名为例，说明域名的注册过程。

操作步骤如下：

① 登录中国万网（http://www.net.cn），在查询域名的查询文本框中输入要申请的域名。例如查询一个 haoxinwen 的域名，如图 10-1 所示。



图 10-1 域名查询

② 单击【查询】按钮，出现查询结果页面，如图 10-2 所示。







图 10-2　域名查询结果

③ 在域名的查询结果中选择需要注册的域名，单击【单个注册】按钮，将进入到申请年限和产品种类的选择页面，如图 10-3 所示。



图 10-3　年限和产品选择



④ 选择年限及产品后，单击【提交】按钮，进入信息填写页面，在域名注册的表单中填写域名的注册资料。这些域名的注册资料可用于域名续费等其他业务，需要填写准确的注册资料，如图 10-4 所示。

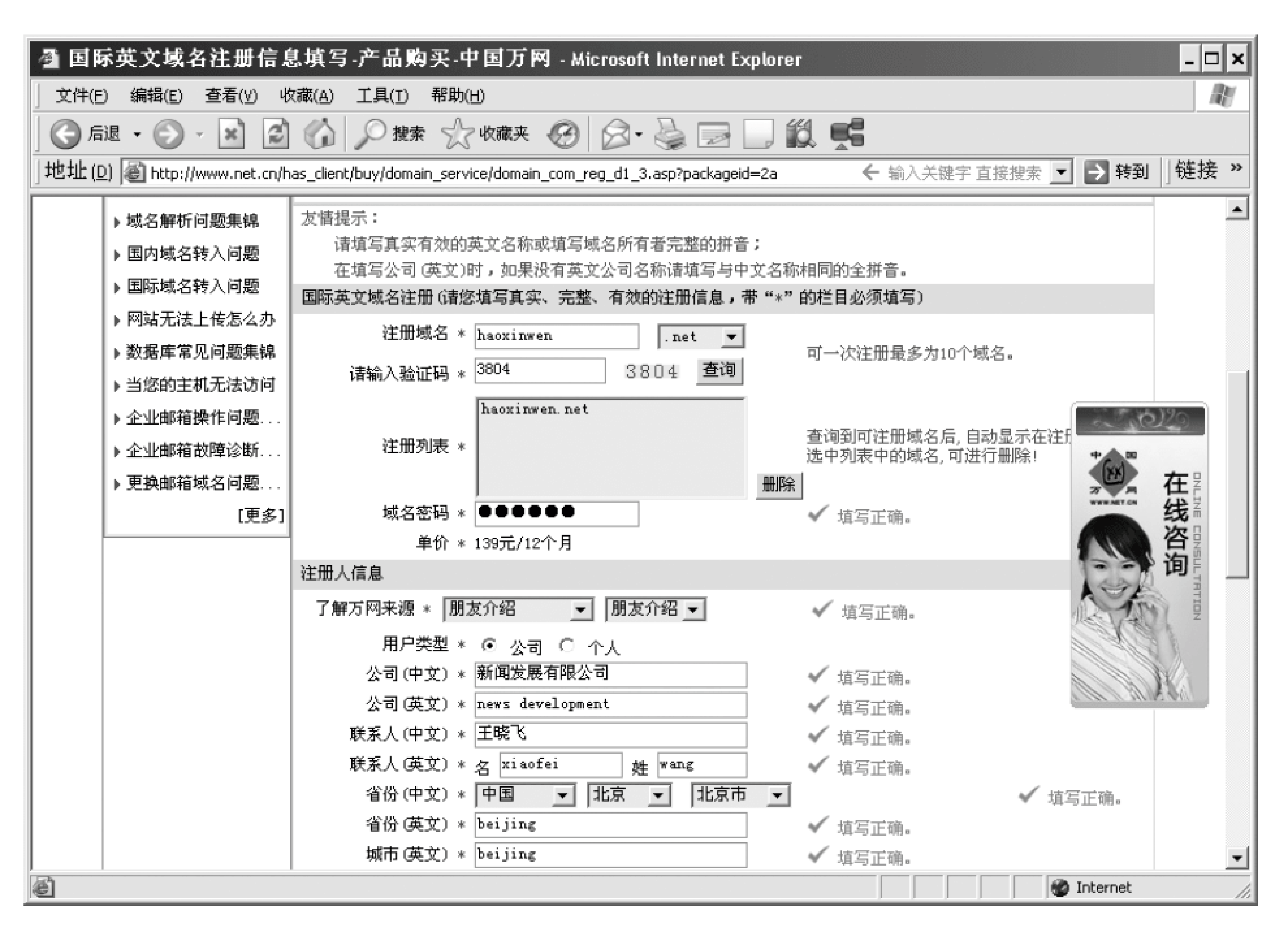


图 10-4　信息填写

⑤ 填写完成后提交这个域名的注册信息，会进入注册成功页面。提交成功后，还需要按照该域名服务商的交费规定进行交费。域名的购买一般都可以在线支付，选择一个支付方式进行在线支付，如图 10-5 所示。域名完成注册并交费后，即可使用这个域名指向"新闻类别网站"(将域名指向网站的任务这里还不能做，因为网站还没有上传，在完成后续的任务后，即可实现此任务。请参考后续内容)。

(4) 域名解析

为"新闻类别网站"进行域名解析。

小贴士

大多数域名注册网站都提供域名解析功能，进入域名解析页面一般都需要先进行登录。

登录中国万网自助域名解析系统，如图 10-6 所示。

在域名服务商的域名解析页面上用域名和域名管理口令登录域名管理系统，进入域名管理页面后，单击【域名解析】按钮可以进入域名解析网页，增加域名解析。一个域名要实现 HTTP 服务，则要解析到一个 Web 服务器的一个固定 IP 上，让域名映射到该 Web 服务器。域名解析填写正确后，单击【提交】按钮，即可生效。

任务 2：申请服务器空间

企业可能采用的服务器形式有三种，如前面的知识储备所述。下面以虚拟主机和服务器托管方式为主，以自建服务器为辅，分别讲述完成"新闻类别网站"的服务器空间申请任务的操作方法。







图 10-5　在线支付



图 10-6　域名解析

子任务 1　申请虚拟主机

子任务 2　申请服务器托管

子任务 3　自建服务器

子任务 1：申请虚拟主机

为“新闻类别网站”申请虚拟主机。



注意

这里只给出工作步骤，不给出工作结果，请读者自行在网上申请虚拟主机。

(1) 确定需要的服务器空间类型和大小

小贴士

不同服务器空间的主要区别是支持网站程序和支持数据库的不同。

常用的服务器空间可能分别支持下面这些不同的网站程序。

- ASP：使用 Windows 系统和 IIS 服务器。
- PHP：使用 Linux 系统或 Windows 系统，使用 Apache 网站服务器。
- .NET：使用 Windows 系统和 IIS 服务器。
- JSP：使用 Windows 系统和 Java 的网站服务器。

不同的服务器空间可能支持不同的数据库，常用数据库有以下几种。

- Access：常用于 ASP 网站。
- SQL Server：常用于 ASP 网站或.NET 网站。
- MySQL 数据库：常用于 PHP 网站或 JSP 网站。
- Oracle 数据库：常用于 JSP 网站。

在注册服务器空间时，需要选择支持自己网站程序与数据库的服务器空间。例如，本教材中“新闻类别网站”采用的开发技术是 JSP，数据库管理系统是 SQL Server，需要选择支持 JSP 和 SQL Server 的服务器空间。同时，需要注意所申请的服务器空间大小，100MB 的空间可存放一般规模的网站。如图 10-7 所示是中国万网的虚拟主机申请页面。



图 10-7 虚拟主机类型

(2) 选择提供虚拟主机的服务商

请读者多了解一些提供虚拟主机的服务商，咨询交费规定和提供的服务等相关内容，选择较适合自己的服务商。



📖 **小贴士**

比较著名的有中国万网(www.net.cn)、新网互联(www.dns.com.cn)、商务中国(www.bizcn.com)、35 互联(www.35.com)、时代互联(www.now.cn)、中国 E 动网(www.edong.com)等。

(3) 联系服务商

与服务商进行沟通,咨询一些与申请虚拟主机有关的事宜。

📖 **小贴士**

几乎所有的虚拟主机服务商都提供在线 QQ 咨询服务,咨询起来很方便。

(4) 申请服务器空间

进入选好的服务商网站的虚拟主机申请页面,按照提示进行申请服务器空间的操作。

📖 **小贴士**

不同服务商的申请方法和流程可能会有不同,大体流程是:在服务商网站上注册会员→选好需要的空间→填写相关信息→下订单→选择支付方式(在线银行支付或邮政支付等)并支付→服务商确认收到钱后开通空间。

子任务 2:申请服务器托管

申请服务器托管的方法与前面虚拟主机的申请类似,同样需要选择服务商,上网申请、在线付费等操作,这里就不再赘述。请读者自行申请服务器托管。

📖 **小贴士**

服务器托管,可以采用主机托管和主机租用两种方式。

主机托管是用户自备服务器硬件,自己安装软件,由服务提供商负责该主机能够连接到网络上及保证该主机能够正常运行。中国万网就是一个不错的服务器托管公司,单击万网首页导航条的独享主机,选择其中的主机托管,如图 10-8 所示。



图 10-8　主机托管



主机租用是由服务提供商提供硬件，负责基本软件的安装、配置，负责服务器上基本服务功能的正常运行，用户独享服务器的资源，并可运行自行开发的程序。单击万网的独享主机，可以选择租用什么样的服务器，如图 10-9 所示。



图 10-9 主机租用

子任务 3：自建服务器

一般的中小型企业不建议采用此种方式，在此不再细述。有兴趣的读者可自行查找相关资料（当然这种方式对读者的自身能力要求也较高）。

10.1.3 相关知识拓展

1. 域名

按照管理机构与域名后缀的不同，网站的域名可分为国内域名与国际域名两种。

- 国内域名：以.cn 作为域名后缀的域名就是国内域名。如 www.sina.com.cn 是一个国内域名，而 www.sina.com 则是一个国际域名。国内域名是由中国互联网信息中心（http://www.cnnic.net）统一管理的。中国互联网信息中心授权一些网络公司经营和管理国内域名。
- 国际域名：各个国家都大量使用没有国家字段的域名，被称作国际域名。如 www.baidu.com 就是一个国际域名。域名起源于美国，美国的域名省略了国家字段。国际域名是由美国统一管理的。常见的以.com、.net、.org 结尾的域名都是国际域名。国内的一些网络公司代理国际域名。

域名的倒数第二个字段或国际域名的最后一个字段代表域名的种类。表 10-1 为一些



常见的域名后缀类型。

表 10-1　常用的域名字段

字　段	类　　型	字　段	类　　型
. com	商业机构域名	. cc	商业公司域名
. net	网络服务机构域名	. info	信息和信息服务机构
. org	非营利性组织	. tv	电视媒体域名
. gov	政府机构域名	. biz	商业机构域名
. edu	教育机构域名		

对于. gov 政府域名、. edu 教育域名等类型的域名，要求申请单位必须具备相关资质，申请和注册时需要提供有效的证明材料。

以下是一些经常采用的域名选择方法。

- 直接使用实际机构的简称作为域名，例如 www. cctv. com、www. cnnic. com。
- 使用机构的英文单词作为域名，例如 www. people. com. cn、www. chinachannel. com。
- 使用机构的汉语拼音作为域名，例如 www. alibaba. com、www. baidu. com。
- 使用品牌名称作为域名，例如 www. lenovo. com、www. hasee. com。
- 使用数字作为域名，例如 www. 163. com、www. 35. com。
- 使用汉语拼音与单词或数字的组合作为域名，例如 www. hao123. com、www. e21. com. cn。

2. 虚拟主机、服务器托管、自建服务器比较

(1) 虚拟主机的特点

- 建站方便迅速：现在主要的服务商都已经实现了业务流程的电子商务化，选择适合自己需要的虚拟主机，在线付款之后马上就可开通，非常方便快捷。
- 服务器管理简单：如果采用虚拟主机方案，则诸如软件配置、防病毒、防攻击等安全措施都由专业服务商提供，可大大简化服务器管理的复杂性。
- 成本相对较低：相对于购买独立服务器，虚拟主机可以大大降低小型企业在网站人力、财力等方面的投入成本，并可获得专业的维护和较好的性能。

因此，虚拟主机空间适合于网站内容比较少，功能不是太复杂，访问量也不是很大的企业网站。

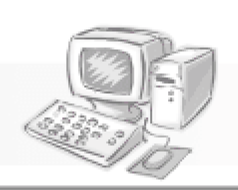
(2) 服务器托管的特点

服务器托管采用的是专门服务器，通过这种方式建立的企业网站，具有更强的功能和更大的灵活性，可自主决定运行的系统和从事的业务。不过，这种方式要求企业具有一定的经济实力，能够承担服务器组建、线路租用以及后期维护费用等。

对于中小型企业而言，如果企业网站访问量大，有适合企业的运作机制，但又无法投入更多的人力做后期维护，那么服务器托管的方式将是首选。

(3) 自建服务器的特点

若企业自建服务器，则需要有水平较高的专业技术人员，还得投入较大的资金购置较好



性能的服务器,并且日常维护工作量也非常大。这种方式完整的构建流程包括:申请互联网接入线路、申请网站域名、购置网站服务器、安排专门的技术人员管理与维护,可以说其投入成本是相当大的。

虽然自建服务器方式在投入成本方面大大高于虚拟主机以及服务器托管,但其体现的特点也是十分显著的。企业自己组建网站服务器,一方面可以根据企业实际情况配置服务器,安装适合企业运作的管理软件,更快地获得市场商机、加强与客户的沟通;另一方面可以通过企业网站服务器建立内部局域网,加强员工之间的交流,也方便了企业对员工工作的监控与管理。

目前,成熟的大中型企业均可考虑采用此类网站建设方式。其完整的网站管理环境可以全面提升网站运营质量,让企业更好地利用互联网为自己服务,同时也可以加强企业自身的建设,全面实现企业的信息化。

10.1.4 经验总结

域名是企业在网络上的商标,为网站起一个好的域名非常重要。一个好的域名能够让人很容易地联想到该企业或企业所从事的业务,这对宣传推广网站是非常有利的。

在选取域名时,主要考虑希望域名拥有什么样的含义,是要包含企业名字,还是要包含企业商标,或是要包含产品类型,以及是否要将域名印制出来进行广告宣传等。一般来说,域名要便于人们记忆,而且要具有实际意义。没有特殊含义的或是太长的不便于记忆的域名是不宜使用的。如果企业已经拥有公认的商标或者名字,可以使用它们来作为域名。

注册域名时,需要选择服务好的域名代理商进行注册。在续费、域名解析等后期服务中,好的域名服务商可以提供更好的域名服务。

选择何种服务器方式构建网站,则要根据企业自己的实际情况,需要考虑企业的互联网应用和企业人力、财力、物力等各方面的支持。

服务器空间可以申请付费空间,也可以申请免费的空间。在互联网上有很多公司提供免费空间给个人使用,同时提供一个免费的域名。但这种免费的域名一般是二级域名或带免费域名机构相应信息的一个链接目录。免费空间通常都比较小,功能也很少,一般只支持静态的HTML文件,而且服务没有必要的保证。但是,免费空间只需要向服务商的网站提供一些个人信息就可以得到空间,这对于刚开始学习网页制作的人们来说是非常合适的。具体操作方法是在搜索引擎上搜索免费主页空间,然后根据要求填写相关的个人注册信息即可。

10.2 发布网站

10.2.1 任务描述

发布“新闻类别网站”。

10.2.2 任务实现

发布网站又称为上传网站,是把做好的网页文件发布到服务器空间。发布网站的方法



有多种，下面分别使用 Dreamweaver、LeapFTP 和 IE 浏览器三种工具来实现网站的上传。

任务 1　使用 Dreamweaver 上传网站

任务 2　使用 LeapFTP 上传网站

任务 3　使用 IE 浏览器上传网站

任务 1：使用 Dreamweaver 上传网站

Dreamweaver 提供上传网站的功能，这需要先在 Dreamweaver 中建立一个站点。在前面章节中就创建过本地站点，现在要为该站点设置远程站点的相关信息。具体操作步骤如下：

（1）选择【站点】菜单中的【管理站点】命令，打开【管理站点】对话框，选中要上传的站点，如图 10-10 所示。然后单击【编辑】按钮。



图 10-10　【管理站点】对话框

（2）打开【站点定义】对话框后，在【高级】选项卡的【分类】列表框中选中【远程信息】。在【访问】下拉列表框中设置远程服务器上传文件时的访问类型，一般情况下为 FTP 服务，如图 10-11 所示。

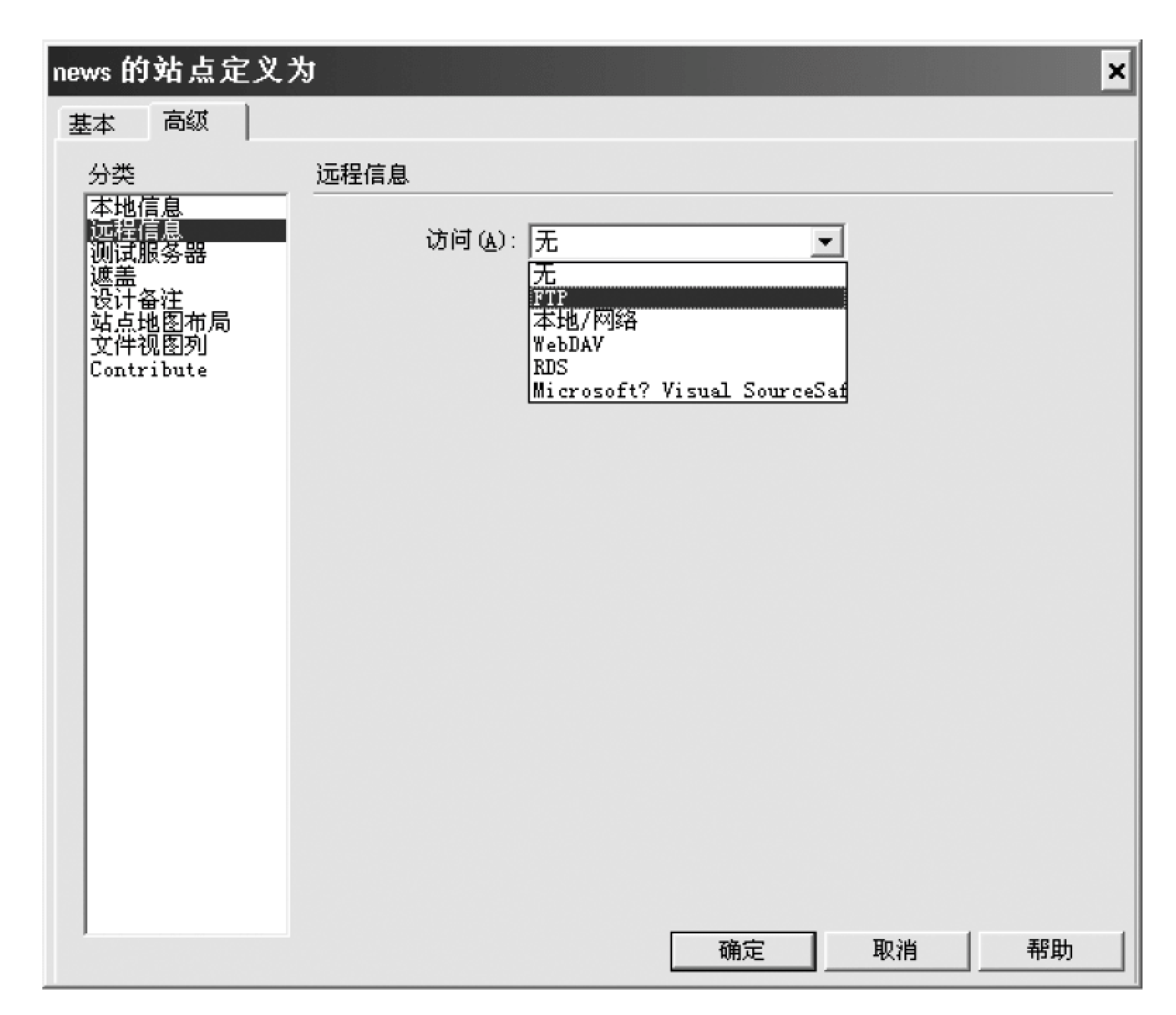


图 10-11　设置站点的访问类型

小贴士

【访问】下拉列表框中有 6 个选项，各选项的含义如下：

- 无：表示不上传网页文件。
- FTP：表示以 FTP 传输方式将网页文件上传到远程服务器站点。



- 本地/网络：表示将文件传送到本地局域网的共享文件夹下。
- WebDAV：使用 Dreamweaver 连接 WebDAV 服务器。
- RDS：使用远程发展服务进行资源管理。
- SourceSafe 数据库：表示利用 Dreamweaver 将网页与 SourceSafe 数据库进行链接，以使浏览者能访问该数据库。

（3）从【访问】下拉列表中选择 FTP，打开网站定义对话框中的 Web 服务器信息页，如图 10-12 所示。

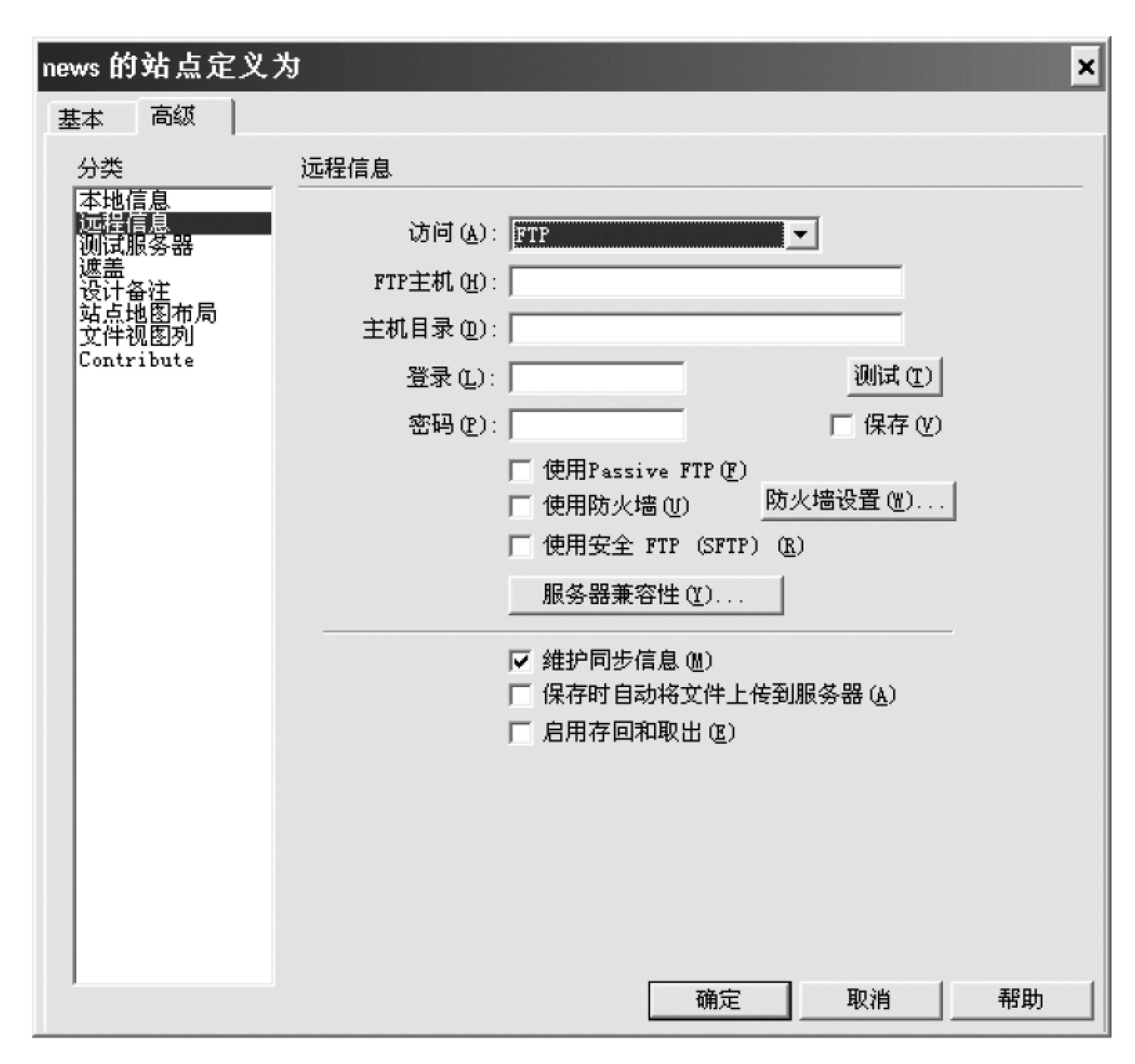


图 10-12　设置站点的远程信息

执行如下设置信息的步骤：

① 在【FTP 主机】文本框中输入远程服务器的主机名称或 IP 地址。

② 在【登录】和【密码】文本框中输入前面申请空间时得到的用户名和密码。

③ 其他选项选择默认值，单击【确定】按钮，返回到【管理站点】对话框，单击【完成】按钮结束设置。这样就可以使 Dreamweaver 能够以 FTP 的方式把本地站点的文件上传到服务器上。

（4）在【文件】面板中，单击【展开以显示本地和远端站点】按钮，则在站点管理窗口中显示本地和远程的文件，窗口左侧是远端站点视图，右侧是本地站点视图，如图 10-13 所示。

（5）单击【连接远端主机】按钮，将 Dreamweaver 与远程网站接通。接通后，网站管理窗口的远程网站窗口中将自动显示远程网站文件目录。

（6）在【本地文件】窗口中选中要上传的文件，这里选择站点的所有文件，然后单击【上传文件】按钮，就开始上传文件了。网站文件上传完成后，【远端站点】窗口将列出远端站点的



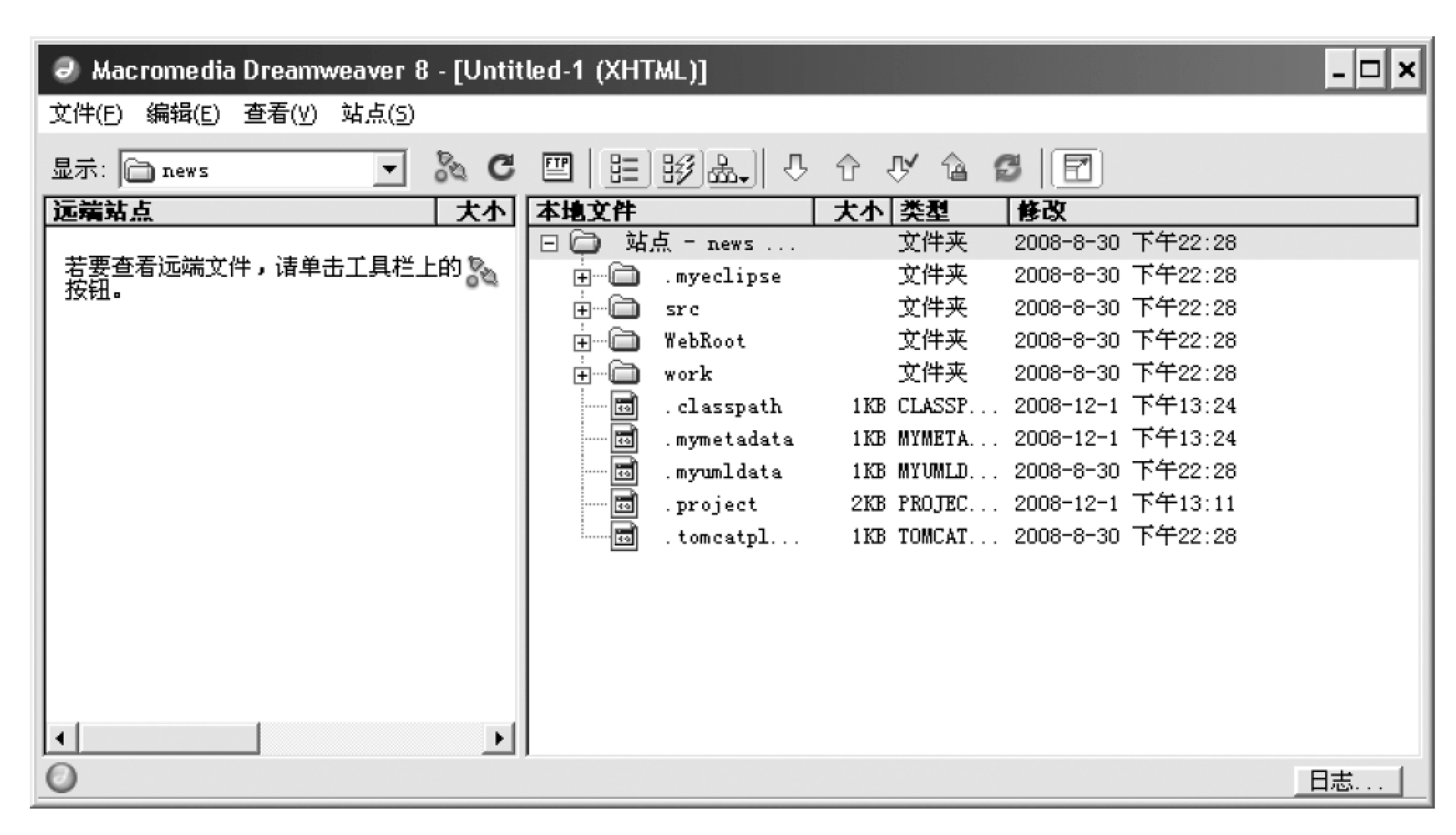


图 10-13　站点管理窗口

文件列表，如图 10-14 所示。此时在左右两个窗口中可以看到一模一样的文件和文件夹。

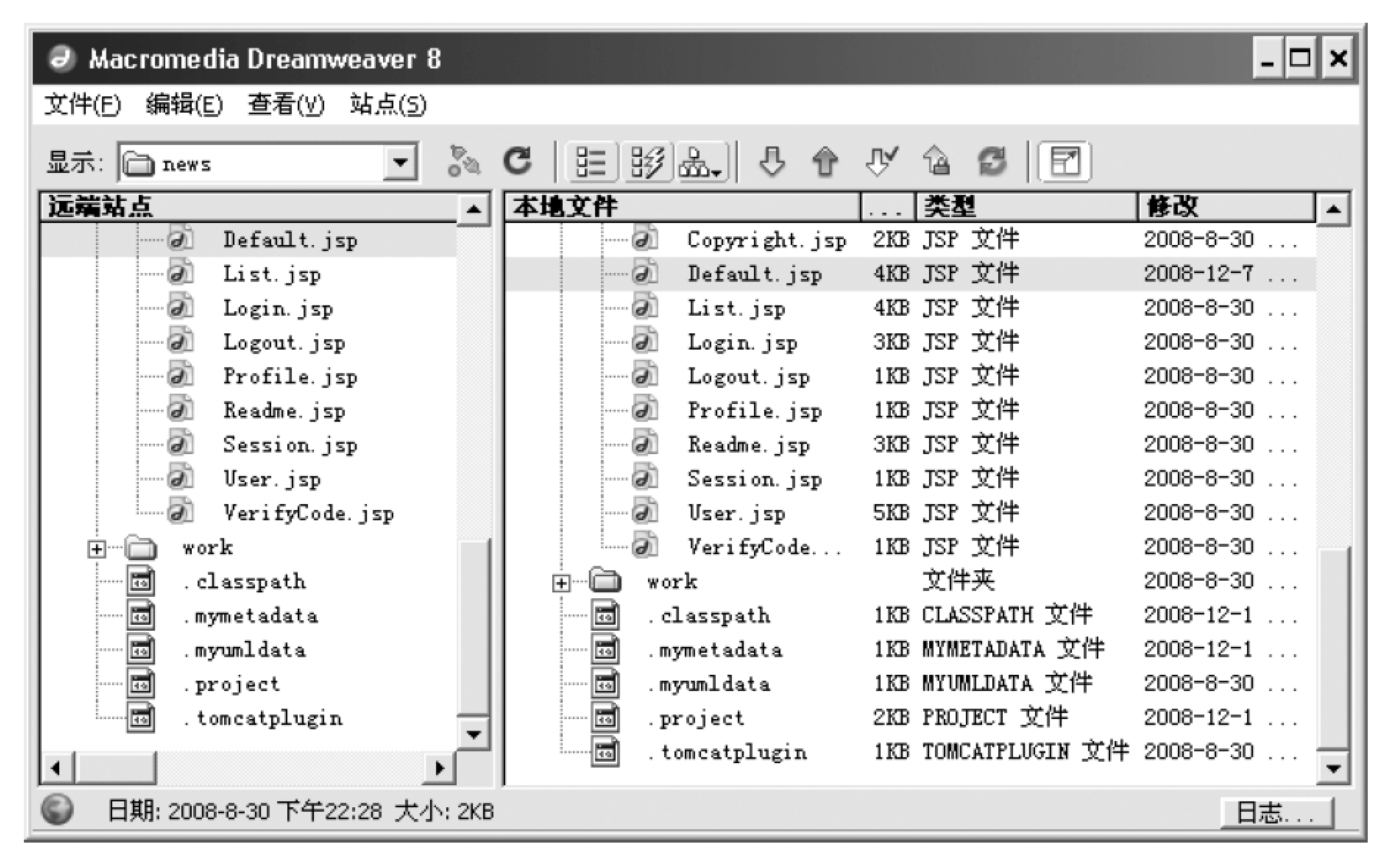


图 10-14　网站上传页面

任务 2：使用 LeapFTP 上传网站

使用 FTP 软件上传网站是最常用的上传方式，FTP 软件有 CuteFTP、LeapFTP、FlashFXP 等。其中 LeapFTP 是一款很受欢迎的 FTP 软件，方便实用，容易上手。

从网上下载 LeapFTP，得到的是压缩包。将已经下载的压缩包解压缩，可以看见如图 10-15所示的一些文件。

使用 LeapFTP 上传网站的具体步骤如下：

(1) 双击运行 LeapFTP.exe 文件，打开 LeapFTP 窗口，如图 10-16 所示。






图 10-15　LeapFTP 的文件

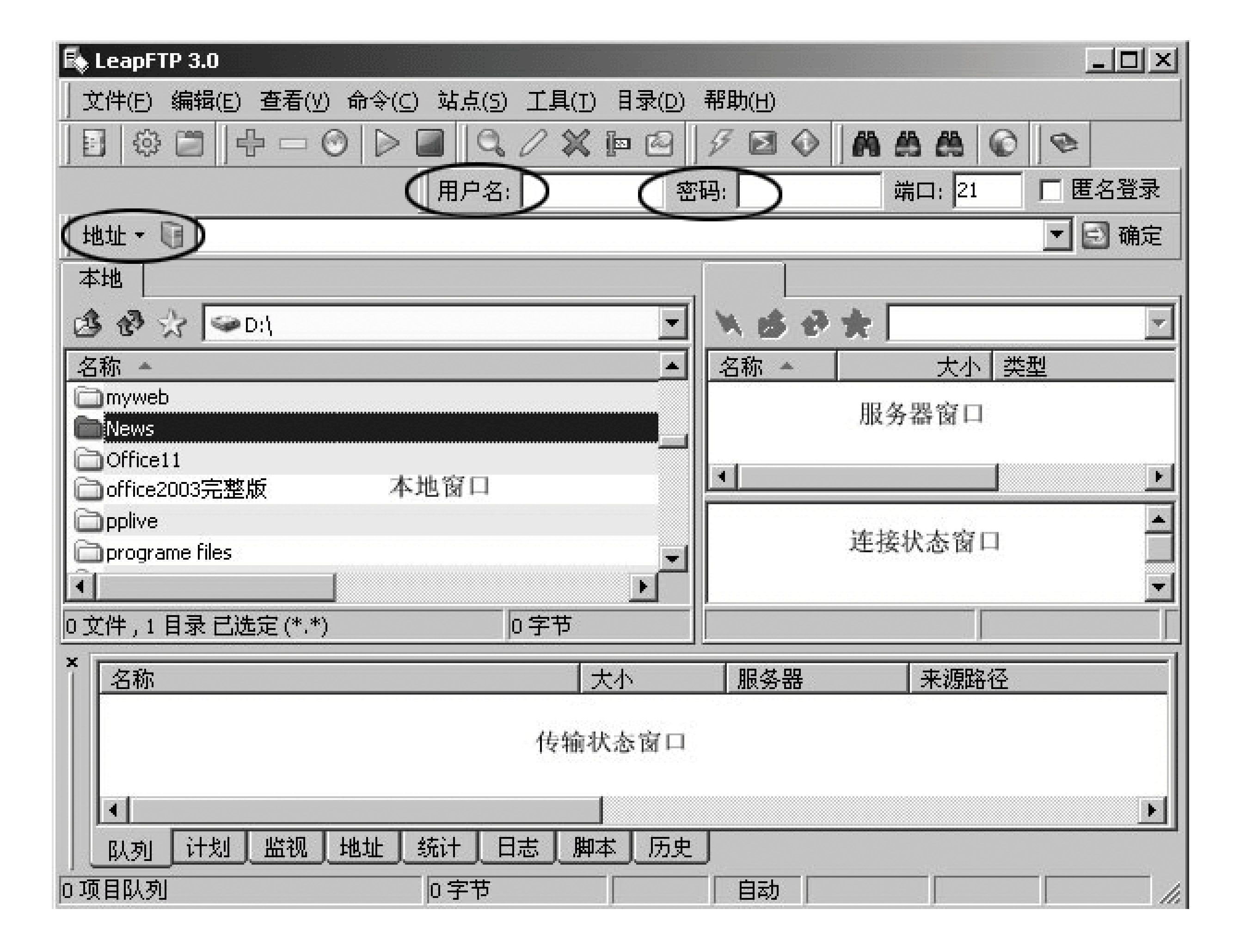


图 10-16　LeapFTP 主界面

小贴士

LeapFTP 主界面的窗口有以下几种。

- 本地窗口：显示本地硬盘空间的内容，上传时可以通过下拉菜单选择网站的本地目录。
- 服务器窗口：显示 FTP 服务器的内容。如果连接上服务器，就可以看见服务器上有什么目录，有什么文件，窗口上面是当前所在位置路径。
- 传输状态窗口：显示当前上传或者下载的文件列表，当某一文档传输成功后，它会自动从此列表中消失。
- 连接状态窗口：FTP 命令行状态显示区，主要显示在连接过程中的验证信息。
- 服务器地址栏：用于填写远程 FTP 服务器的地址，可以是域名，也可以是 IP 地址。
- 用户名和密码栏：填写在远程 FTP 上的账户名和密码，端口默认是 21。

(2) 连接服务器。

在主界面的相应位置输入登录服务器的各项设置信息：

① 在服务器地址栏中输入 FTP 服务器地址或域名。

② 在用户名栏中输入申请服务器空间时的用户名。

③ 在密码栏中输入申请服务器空间时的密码。

④ 端口使用默认的 21。

⑤ 去掉匿名登录前的对钩，有的空间不支持匿名登录。

设置完成后，单击【确定】按钮，开始连接服务器。连接成功后，服务器窗口中会显示所连接到的服务器上的目录。接下来就可以上传文件了，如图 10-17 所示。

(3) 上传网站。

从本地窗口选定要上传的网站文件夹或文件，用鼠标拖至远程服务器窗口，即开始上传工作。等上传结束后，就可以在服务器窗口中看到上传的网站，与本地窗口的网站内容一致，如图 10-18 所示。

上传是最基本的站点操作，另一个常用操作是下载。如果想从服务器下载网站或某个文件，可以选中要下载的内容，在右击快捷菜单中选择【下载】命令即可。



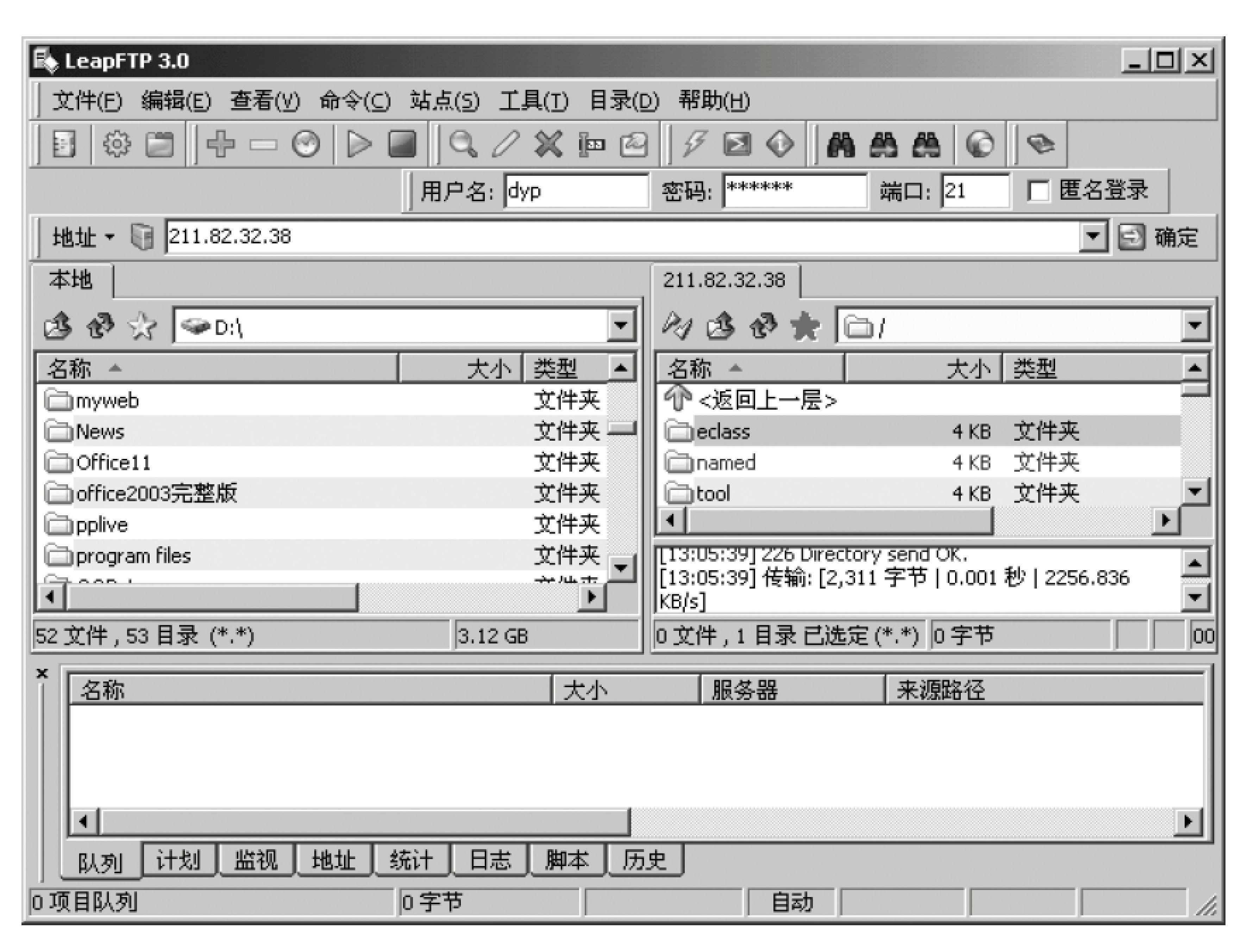


图 10-17　连接服务器



图 10-18　上传网站

任务 3：使用浏览器上传网站

如果手头没有可用的 FTP 软件，也可以直接用浏览器打开 FTP 服务器。其实就是在浏览器中建立 FTP 连接，将上传的操作转化为资源管理器中的复制和粘贴的操作。

具体操作步骤如下：

(1) 在 IE 浏览器的【地址】栏中输入"FTP：//"，后面输入 FTP 服务器的地址，这里的



FTP服务器地址就是已申请的服务器空间的域名。

(2) 连接时会弹出【登录身份】对话框，如图10-19所示。输入用户名和密码，单击【登录】按钮。

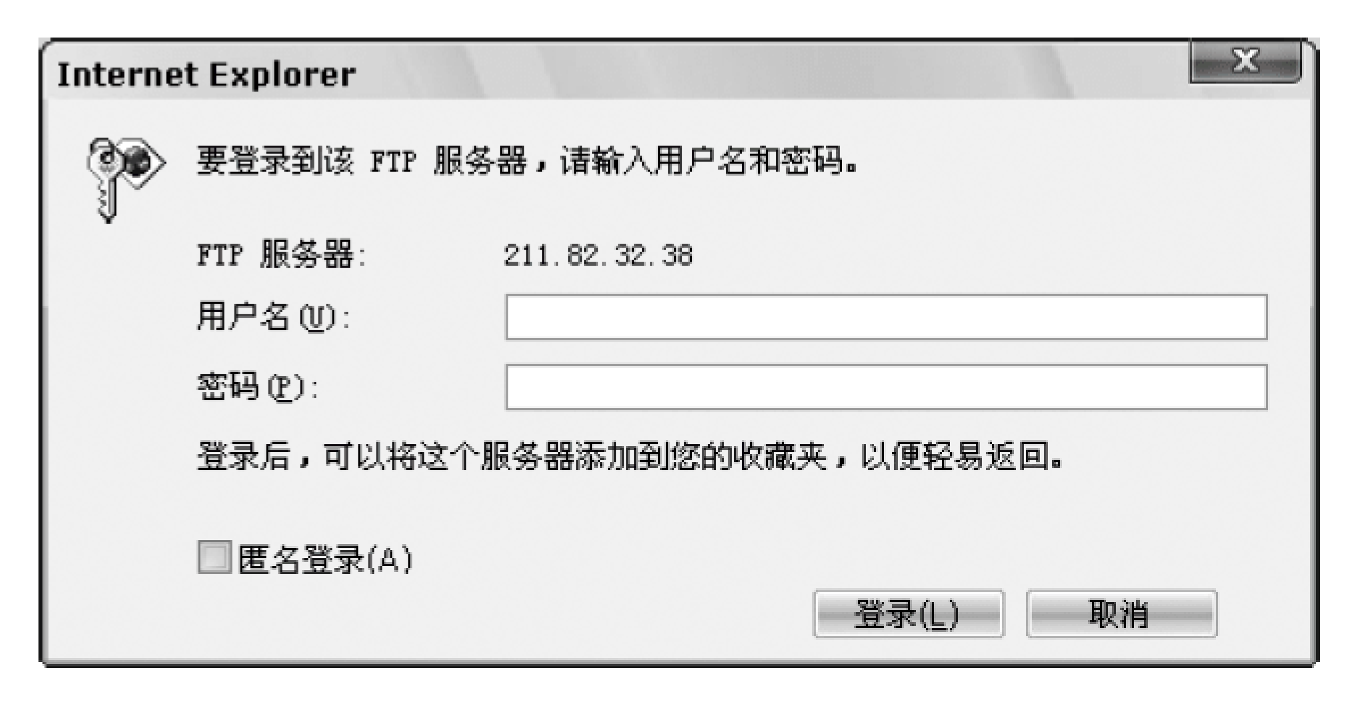


图10-19 【登录身份】对话框

(3) 登录后可看到对应服务器空间中网站里的内容。此时就可以像使用自己的计算机中的本地文件夹一样把网页文件复制、粘贴过去。

10.2.3 相关知识拓展

1. FTP协议

FTP(File Transfer Protocal)的含义是文件传输协议。一般来说，访问网络的首要目的就是实现信息共享，而文件传输是信息共享的重要内容之一。Internet上早期实现传输文件，并不是一件容易的事。Internet是一个非常复杂的计算机环境，有PC，有工作站，有MAC(苹果计算机)，有大型机，而这些计算机可能运行不同的操作系统，有运行UNIX的服务器，也有运行DOS、Windows的PC和运行MacOS的苹果机等，而各种操作系统之间的文件交流问题，需要建立一个统一的文件传输协议，这就是所谓的FTP。基于不同的操作系统有不同的FTP应用程序，而所有这些应用程序都遵守同一种协议，这样用户就可以把自己的文件传送给别人，或者从其他的用户环境中获得文件。

FTP采用"客户机/服务器"方式，支持FTP协议的服务器就是FTP服务器，用户要在自己的本地计算机上安装FTP客户机程序。用户可通过客户机程序从远程服务器下载文件，也可以上传文件到远程服务器。例如，当用户启动FTP从远程计算机复制文件时，本地FTP客户程序向FTP服务器提出复制文件的请求，远程FTP服务器程序响应请求，把指定的文件传送到用户的计算机上，客户机程序代表用户接收文件，将其存放在用户目录中。

正如其名，FTP的主要作用就是在Internet上传输文件，并提供登录、目录查询、文件操作及其他控制功能。如果要建立或更新自己的网站，一般使用FTP上传、下载来完成网站的发布、管理和维护。

2. 设置站点远程信息的补充

参见图10-13，设置站点的远程信息时除了前述的基本设置外，还可以设置一些其他信息。

在【主机目录】文本框中可以设置服务器中保存文档的目录，此目录路径可以从Web服务器管理员那里得到，假如没有设置目录，这部分可以为空。





选中【使用 Passive FTP】复选框，则此 FTP 连接方式可以为被动式连接。在通常的 FTP 传输过程中，一台计算机（客户机）先与其他计算机（服务器）建立连接，然后请求发送数据，而防火墙保护（Firewall-protected）的服务器不允许建立这种初始连接，因此无法传送数据。被动式 FTP 是通过本地的软件而不是通过服务器来建立连接的。大多数防火墙配置使用被动式 FTP，用户可以询问网络管理员来决定是否需要该设置。

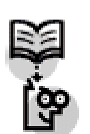
选中【使用防火墙】复选框，可以设置是否使用防火墙。许多公司使用防火墙这个安全特性来防止对其内部文档的不期望的访问。防火墙有多种类型，每一种都有多重安全设置。例如，有些防火墙允许公司内部人员自由地互相移动文档，而有些公司可能不允许 Java 或者 ActiveX 控件通过防火墙移动。

单击【测试】按钮，可以测试输入的 FTP 主机地址及其用户名和密码是否正确，如果连接成功，会弹出一个连接成功确认对话框。

10.2.4　经验总结

成功申请空间和域名之后，接下来就是将制作好的网站上传到服务器上以便让别人访问。上传都是通过 FTP 完成的，但是可以采用三种不同的方式：使用 FTP 软件上传、使用网页制作工具上传和使用浏览器上传，其中，使用 FTP 软件进行上传是最基本也是最常用的上传方式。

10.3　宣 传 网 站


知识储备

网站发布到 Internet 上之后，要通过各种方式宣传推广网站，让尽可能多的人以及预定的目标访问者了解网站，提高网站的点击率，这个工作称为网站推广。网站推广有免费推广和付费推广两种方式。

10.3.1　任务描述

宣传推广“新闻类别网站”。

10.3.2　任务实现

采用登录搜索引擎的方式，把“新闻类别网站”添加到各种搜索引擎。

任务 1　利用免费搜索引擎推广网站

任务 2　利用付费搜索引擎推广网站

任务 1：利用免费搜索引擎推广网站

（1）百度搜索引擎的申请

打开 http://www.baidu.com/search/url_submit.htm 页面，在输入框内输入网址与



验证码，单击【提交网站】按钮，完成注册，如图 10-20 所示。

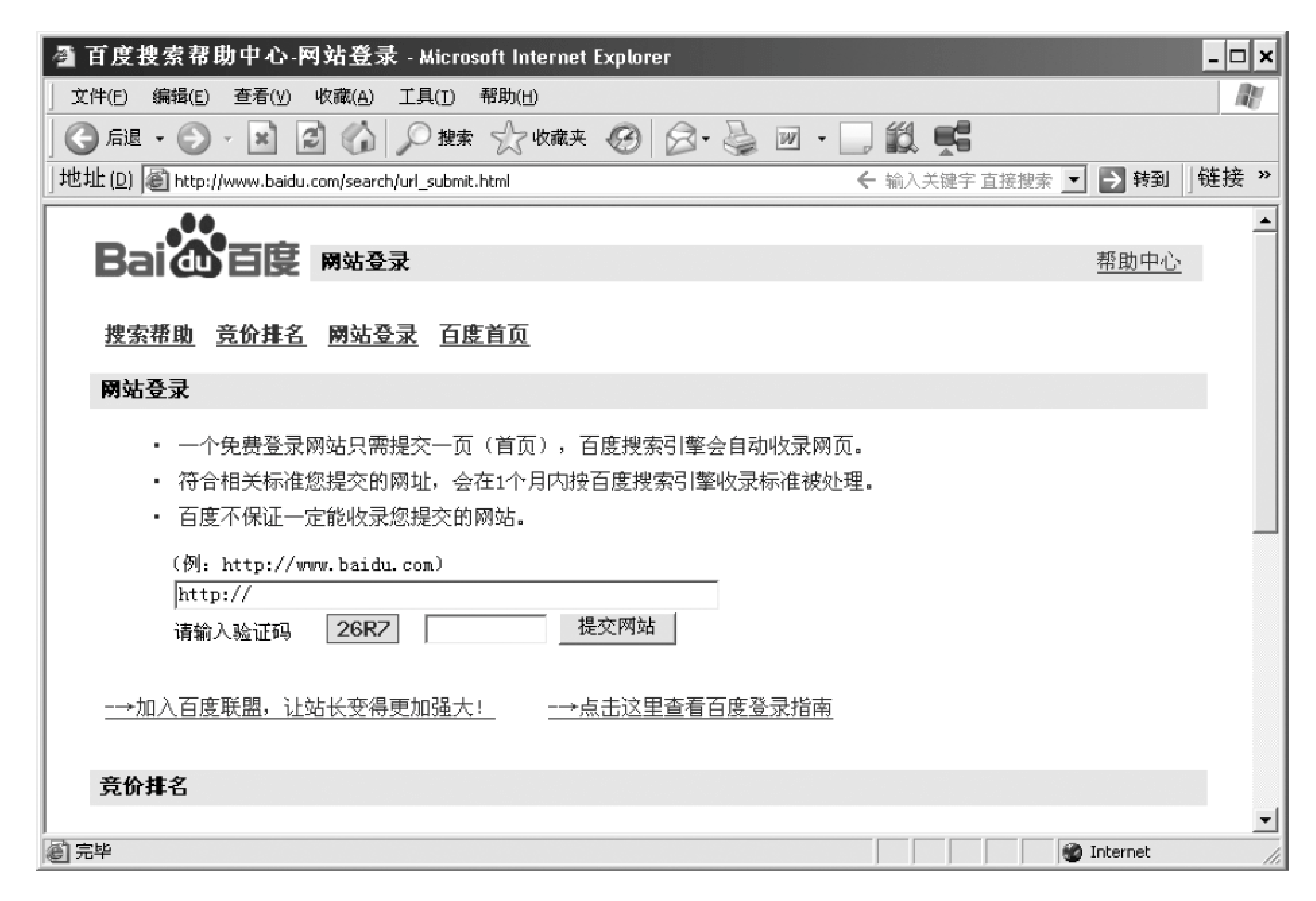


图 10-20　百度搜索引擎登录

（2）Google 搜索引擎的申请

打开 http://www.google.com/addurl 页面，输入网址和评论，单击【添加网址】按钮，就可以成功提交，如图 10-21 所示。

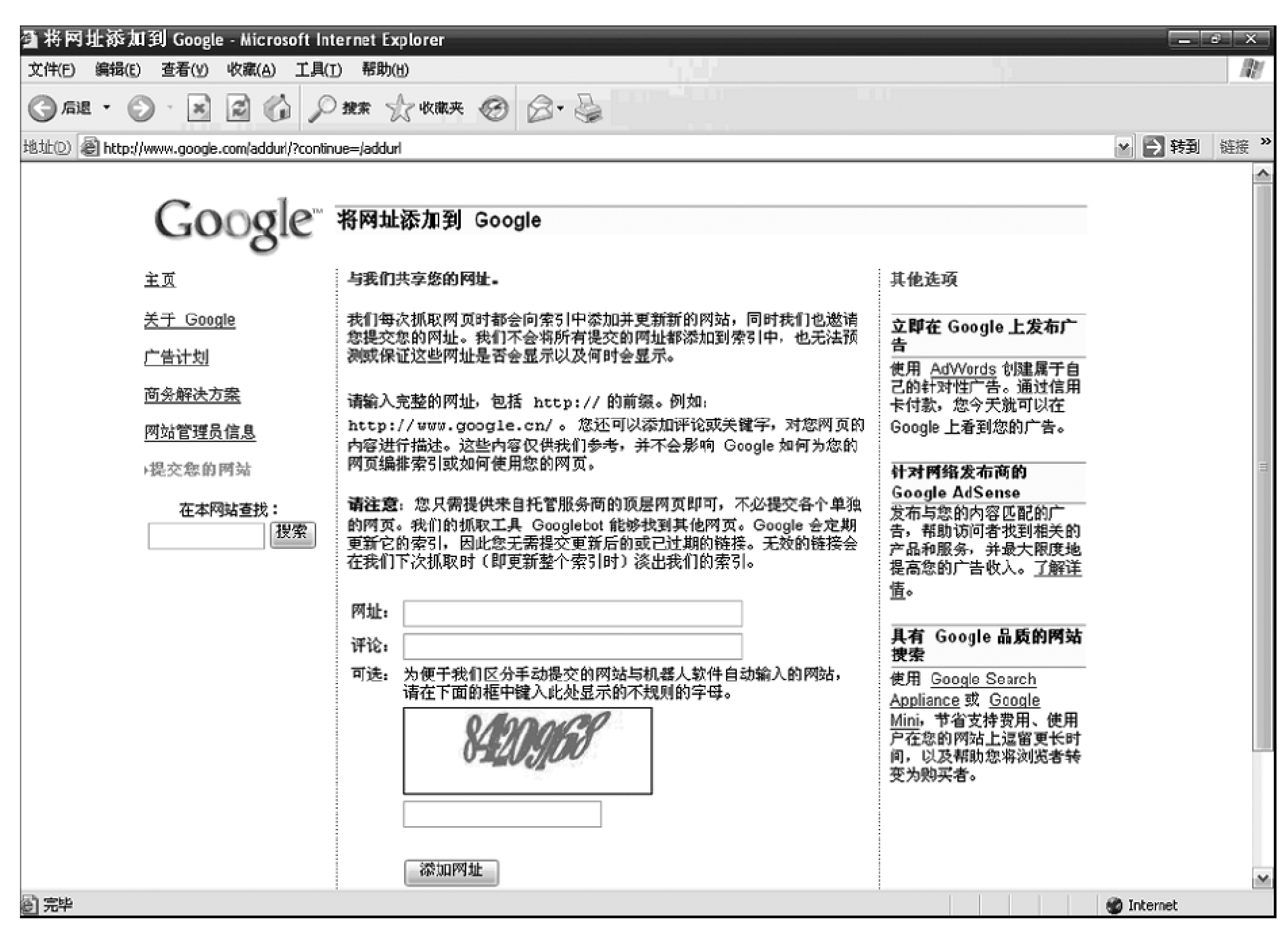


图 10-21　Google 搜索引擎登录



📖 **小贴士**

对于中文网站，目前要注册的两个搜索引擎是百度和 Google。Google 虽然在世界上几乎成为搜索的代名词，但是百度以其专注于中文世界，内容更新快、断词顺序合理以及市场宣传等优势，在市场上与“Google 中文”并驾齐驱。百度和 Google 的登录是免费的，一般只需将站点首页地址提交，它们就会自动前来抓取全部内容。

提交网址后，站点会在 1～2 天内出现在百度搜索引擎中，而出现在 Google 的时间不是很确定，因为 Google 刷新数据库的周期长，所以最长可能需要一个月的时间，所提交的站点才能在 Google 中查找到。

当站点有更新时，应该尽量把更新的页面手动递交到上述搜索引擎，以提醒搜索机器人回访，从而更新内容能尽快在这些搜索引擎中被查找到。

一旦站点被收录到百度和 Google 以后，就可以在百度支持的搜狐和网易，以及 Google 支持的新浪页面搜索中查找到。

任务 2：利用付费搜索引擎推广网站

付费搜索引擎推广方式不止一种，下面使用搜索引擎竞价排名方法进行“新闻类别网站”的推广。

(1) 在浏览器的地址栏中输入网址 http://www.baidu.com，打开百度搜索引擎，如图 10-22所示。



图 10-22　百度搜索引擎

(2) 单击首页上的【加入百度推广】按钮，进入百度推广页面，如图 10-23 所示。

(3) 单击【网上申请】按钮，填入用户注册信息，如图 10-24 所示。

(4) 添加关键词，撰写网页标题及描述等信息，如图 10-25 所示。





图 10-23 百度推广

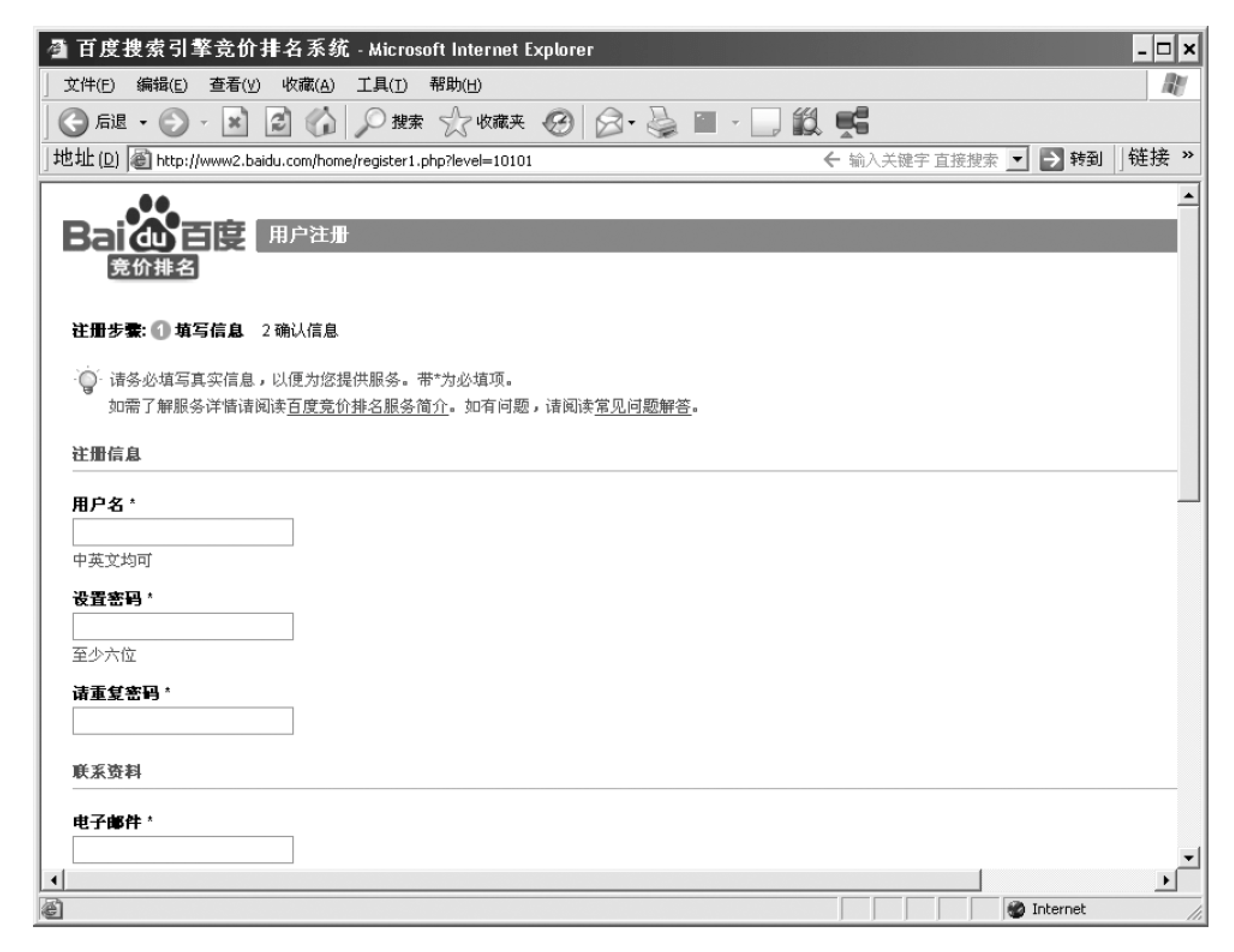


图 10-24 用户注册







图 10-25　添加关键词

（5）选择支付方式，缴纳推广费用。百度在收到款项后，审核用户添加的关键词信息，审核通过会开通账户，按点击效果付费。

10.3.3　相关知识拓展

1. 搜索引擎优化

在做网站宣传时，为了让网站在搜索引擎上的排名靠前，需要对网站进行搜索引擎优化，对网站的标题、关键字等重要信息进行专业分析和设置。

（1）关键字的处理

网页中的关键字主要是为搜索引擎服务的，为了提高网站被搜索引擎搜到的效率，需要设置多个跟网站主题相关的关键字。首先收集与网站或产品有关的关键字，然后将收集到的关键字进行组合，把它们组成常用的词组或短语作为网页关键字。

选择关键字时要从用户的角度考虑，想象一下访问者在尝试查找产品或服务时可能的经历。例如，如果输入“书店”进行查询，无疑将会面对成千上万的查询结果。但是如果查询的是位于北京的新华书店，就会本能地输入“北京新华书店”作为关键字的一部分，而不是使用含糊的关键字“书店”。

（2）meta 和 title 标签

meta 标签是内嵌在网页中的特殊 HTML 标签，包含着有关网页的一些信息。meta 标签的作用是向搜索引擎解释网页是有关哪方面信息的。meta 标签有两个重要的属性：description（网站描述）和 keywords（网站关键词），可以简要地对网页进行描述，并且给出供



搜索引擎使用的关键字。另外一个基本的属性是 title 标签，提示搜索引擎关于网页的主题信息。

meta 和 title 的形式在 HTML 语言中体现为：

```
<head>
    <title>标题</title>
    <meta name=" description" content="在此处编辑当前网页描述文字">
    <meta name=" keywords" content="在此处编辑当前网页的关键字">
    ⋮
</head>
```

meta 标签位于网页的<head>与</head>中。meta 标签的用处很多，目前几乎所有的搜索引擎机器人都通过自动查找 meta 值来给网页分类，作为判断网页内容的基础。

title 标签也很重要，在搜索引擎的搜索中具有非常重要的地位。人们通过搜索引擎搜索出来的是相关网站的链接列表，而显示的网站链接描述就是 title 标签中的标题，所以标题要尽量简洁、明了，让人一看就知道网站是关于什么内容的。

(3) 关键字密度

网页上通常会有数以百计的词语，那么搜索引擎怎样去分辨哪些是描述网站的重要的词语呢？搜索引擎利用特定的算法统计一个页面的字数，哪些重复出现的词或短语被认为比较重要些。关键字数与该页面字数的比例称为关键字密度。为了得到更好的排名，网页的关键字必须在页面中出现若干次，或者出现次数在搜索引擎允许的范围内。

(4) 关键字位置

要考虑把关键字放在网页的什么地方。关键字必须包含在页面文本中，而且必须放在前面。搜索引擎将会专注于网页中某一部分的内容，处于这一关注部分的词语显得比其他部分的词语要重要得多。

① title 和 meta 标签：在前面已经提到，在这两个标签中放置关键字很重要。

② 正文标题(headings)：使用<h1>至<h6>标签进行定义。正文标题为网站访问者指明了哪些是网站中比较重要的内容，在其中出现关键字对于提高网站排名有很大的好处。

③ 超链接文本：除了在导航、网站地图、锚文本中有意识地使用关键字外，还可以人为增加超链接文本，链接到包含有自己网站关键字或相关语义的网站，增加关键字密度。

④ 图片 Alt 属性：搜索引擎网页蜘蛛不能识别图片的内容，因此，在图片属性 Alt 中加入关键词，可以帮助搜索引擎了解图片所代表的含义，有利于排名。

⑤ 顶部：网页顶部的文本，每段开头的内容都特别重要，所以尽量在这些地方把关键字包含进来。

2. 免费推广方式

免费推广网站有很多方法。

(1) 交换友情链接

向一些访问量大或与自己网站目标用户群类似的站点发出建立友情链接的要求，如果



对方同意，就会在对方的网页上得到一个指向我们网站的链接。当然，友情链接要交换，同样在我们的网站的网页上添加指向对方站点的链接。这种形式的宣传效果较好，目标用户群相对明确，可以给自己的网站带来一些流量。

（2）网上发帖

在论坛、博客、留言板上发帖或留言，推广网站。论坛、博客、BBS 是众多网友聚集的地方，到这里来的网友目的相对明确，而且用户分类清晰，所以在这里发布帖子往往能够收到较好的宣传效果。

（3）发送电子邮件

电子邮件作为 Internet 的一项基本功能，已经成为人们重要的联系工具，在进行网站推广时也应充分使用这项功能。收集朋友的 E-mail 地址，通过发送 E-mail 向他们介绍推广网站。邮件主题一定要鲜明，避免被当作垃圾邮件删除。

（4）免费登录搜索引擎

把网站添加到各种搜索引擎，让人们可以在搜索引擎中搜索到这个网站。搜索引擎是大家在网上查找信息的首选工具，拥有巨大的访问量，因此在一些著名的搜索引擎上进行注册是非常有效的网站宣传方法。而且，注册的搜索引擎越多，网站被访问的机会就越多。比较著名的搜索引擎有百度（www. baidu. com）、Google（www. google. com）、新浪（www. sina. com. cn）、搜狐（www. sohu. com）、雅虎（www. yahoo. com. cn）等。

（5）网下宣传

除了在网上进行网站的宣传活动外，在平时的工作、生活中也要宣传网站，比如向亲戚朋友、同学同事介绍，在个人名片上印上网址等。

3. 付费推广方式

付费推广网站有很多方法。

（1）搜索引擎竞价排名

竞价排名是企业在搜索引擎上注册与产品相关的关键词后，网站就会被查找这些产品的潜在客户找到。竞价排名按照给网站带来的潜在客户访问数量进行计费。

（2）分类目录注册

分类目录注册主要是在分类目录合适的类别中进行网站登录，一般需要人工提交网站信息，然后经过分类目录编辑人员审核才能决定是否收录网站。这样就对网站提出了较高的要求，必须符合分类目录的收录原则，而且对网站往往有一定的限制。

一旦被收录分类目录，网站将在一定时期内保持稳定，对于付费分类目录登录，通常需要缴纳年度费用，保持在分类目录中的地位。在高质量的分类目录上登录，对于提高网站在搜索引擎检索结果中的排名有一定价值。但是，由于分类目录收录大量同类网站，并且多数用户更习惯于用搜索引擎直接检索，因此，紧靠分类目录被用户发现的机会相对较小，通常还需要和其他网站推广手段一起使用。

（3）联告

多维定向广告，这个广告产品的特点是按照点击计费，把广告随着匹配结果发布到各大行业、地理门户等中小合作的媒介网站上。



(4) 窄告

产品和联告类似,优势在内容关联定向上做得不错,主要的市场份额集中在北京,北方市场的开发已经相对成熟。其主要与新闻站点合作,公关实力也较强。

(5) 招商信息及行业门户

28商机在线、3158与门户招商频道代理、中国服装网、中国汽车网等,这些招商信息及行业门户主要适合铺设渠道设经销商的企业。

(6) 传统媒体宣传

Internet是一个具有十分诱人发展前景的广告载体,但是传统的方式也不能够完全抛弃。尤其是在目前我国电子商务发展水平不是很高的情况下,要推广企业网站,还应该依赖于传统媒体的宣传。传统媒体的宣传主要可采用户外广告、广播、电视、报纸、新闻发布会、公司广告包装等形式,一般花费较高。

10.3.4 经验总结

宣传网站对于网站的知名度和访问量有着非常大的影响,应该引起网站建设者的高度重视。如同给产品做广告宣传一样,也要通过各种有效方式对网站进行宣传,特别推荐的是搜索引擎注册。同时,要保持网站具有充足的信息和功能。人们在查找信息的时候,总是希望找到一个包括很多重要信息的网站。搜索引擎也是根据网页内容判断网站的质量,而不是从图片、Flash动画上判断。要重视网站内容的更新,因为无论是搜索引擎还是访问者都希望看到比较新的信息。

一般对于个人网站,经常采用免费推广方式。

对于企业网站,可以用付费的方式推广网站。付费推广针对性较好,在较短的时间内即可收到很好的广告收益。

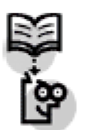
10.4 实　　训

按照本章的知识和工作步骤,对“新闻类别网站”进行宣传推广。

第11章　网站的运行与维护

网站的运行是一个长期的过程，需要坚持不懈地更新、维护和经营网站。要及时地调整和更新网站内容，维护网站长期稳定地运行在 Internet 上，并在瞬息万变的信息社会中抓住更多的网络商机。

11.1　网站更新



知识储备

1. 更新网站内容

网站内容是一个网站的生命力源泉。互联网的最大优势就是信息的实时性，只有快速的反映，准确的报道，才能吸引更多的浏览者。相反，如果一个网站的内容数月如一日，就不会有太多的访问者。同时由于企业的发展状况在不断地变化，网站的内容也就相应需要随之调整，给人常新的感觉，给访问者留下好的印象。

(1) 发布最新消息

随着企业的发展，经常会有动态新闻公布，有新产品、新服务问世，那么就需要在网站及时发布这些新闻、产品、价格及服务信息，如网页图片更新、网页文件资料更新等。

(2) 更新页面设计

对于网站的页面设计，在保证风格统一的前提下，也要经常有更合理的变化，不断增加新的营销创意，提高网站的知名度。

(3) 更换促销广告

网站的一些促销信息随着时间的流逝会失去效用，因此，根据节假日、重大活动、庆典等特定的时间段，一定要更换网站广告或 Flash，陈旧的信息一定要更新。

(4) 访问者信息反馈

对访问者资料信息进行搜集、统计及分析，及时处理他们的投诉或需求信息，并反馈处理结果。

2. 更新网站功能

企业在建站初期，可能出于试探性质的尝试，投资不是很大，但是随着网站项目的实施和经营，可能会给企业带来了较好的形象宣传和经济效益，企业确实看到了网站对于企业的作用，就可能会要求网站增加新的功能，调整更合理的栏目，从而期望带来更多的访问者和



商业机会。

(1) 增加网站功能

网站的主要功能是形象/产品展示和信息发布,根据用户的需求还可以扩充其他一些比较有效的功能,例如网上销售、顾客关系、网上调查、企业论坛、FTP下载等。

(2) 调整栏目版块

为了进一步完善网站结构,可以针对部分栏目版块进行调整,使网站的结构和分类更趋于合理化和明确化,提供更好的服务。

3. 网站升级改版

网站改版指是在已有网站基础上进行整体上的改动,一般有以下两种情况。

(1) 原有网站在设计、网站规范、网站功能等方面都比较完善。企业需要周期性变更网站设计风格,以达到一定的商业目的。

这种网站改版的重点在于设计风格方面,在不脱离企业形象的基础上,根据企业的实际需要,对网站设计风格进行全面更换,使网站给人一种耳目一新的感觉,让访问者感到企业的活力和生气,也从另一个侧面反映了企业的实力。

(2) 原有网站无论从规划、设计,还是内容方面都与企业的运作不相协调,设计风格平庸,未能很好地展现企业形象,功能方面也比较欠缺,网站对企业没有起到实质性作用。企业需要把网站在总体上提升档次,发挥作用。

这种类型的升级改版投资大,项目周期长,需要企业认真筹划。网站的改版最好请专业人士帮助策划,企业熟悉自身的经营特点、运作方式,专业人士会站在专业的角度和消费者的角度分析策划:如何设置导航、如何运用色彩、如何设计网站风格等,使网站既具有美观性,又具有实用性。

11.1.1 任务描述

在"新闻类别网站"的运行中,时常对其进行更新。

11.1.2 任务实现

"新闻类别网站"的更新任务包括三个任务:

任务1 及时更新"新闻类别网站"的内容

任务2 及时更新"新闻类别网站"的功能

任务3 对"新闻类别网站"进行升级改版

任务1:及时更新"新闻类别网站"的内容

经常在网站上发布最新消息,更新页面设计,增加网站创意,更换促销广告,及时处理访问者的信息反馈。

任务2:及时更新"新闻类别网站"的功能

根据需要及时增加"新闻类别网站"的功能,调整相关的栏目版块。

任务3:对"新闻类别网站"进行升级改版

根据需要对"新闻类别网站"进行升级改版。

11.1.3　相关知识拓展

网站一般可分为两种类型：静态网站和动态网站，二者的网页制作语言是不同的。静态网页使用 HTML、XHTML 语言，而动态网页除了使用 HTML 外，还会使用 PHP、ASP、ASP.NET、JSP 等动态网页编程技术。区别静态网站与动态网站的方法通常是看是否采用了数据库的开发模式。静态网站和动态网站的内容更新方式是不同的。

静态网站由直接或间接制作成的 HTML 网页组成，不通过脚本语言、服务器端网页编程技术及数据库开发来实现。这种网页的内容通常是固定的、独立的，页面内容想要变化，必须手工对它进行改动，然后再将修改好的网页上传到服务器。因此，为了不断更新网页内容，就要不断地重复制作 HTML 文档，哪怕一个字符、链接及图片的细微修改和更新都必须将网页重新制作、重新上传到服务器上覆盖原来的页面。当网站内容需要每日或者每周进行更新时，工作量是很大的。

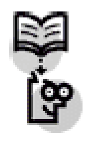
动态网站都使用数据库后台，通过动态网页编程技术（比如 PHP、ASP、ASP.NET、JSP 等）将网站内容数据动态存储到数据库中，用户访问网站时通过读取数据库来动态生成网页内容。网站上主要是用 HTML 实现的网页框架结构，网页的内容大都存储在数据库中。动态网站内容更新时，一般都是通过后台管理系统来完成。登录到后台管理系统后，通过填写并提交表单，可以方便地更新数据库中的信息，这样站点内容也就相当于得到了更新。当网站访问者发出访问请求时，动态生成的网页中就是最新信息。

静态网站与动态网站相比较，二者各有优势与不足。静态网站制作简单快捷，但更新的时候效率很低，而且对于不懂网页编辑的人来讲更新网站也是一大困难。动态网站制作的时候需要花费更多时间，但是站点一旦投入使用，相对来说容易维护，可以把更新站点所需要的时间和工作量减到最小，而且只要会基本计算机操作就可以更新网站内容。

11.1.4　经验总结

一个网站要想长期地吸引住浏览者，必须依靠网站内容的不断更新和网站功能的不断完善来保证。及时更新网站的目的是尽力为浏览者提供最新的信息和最好的服务。静态网站和动态网站的更新方式有所不同，在决定静态网站还是动态网站时，主要考虑网站的功能需求和网站内容的多少。如果网站功能比较简单，内容更新量不是很大，采用静态网站方式会更简单；反之一般要采用动态网站技术来实现。

11.2　维护网站


知识储备

网站维护是指网站运行后期的综合管理和维护，主要包括硬件维护、软件维护、内容维护和安全维护等。

1. 硬件维护

硬件维护主要包括服务器、网络连接设备及其他硬件的维护。要保持所有硬件设备处



于良好状态，维护网络设备不间断地安全运行，并对可能出现的硬件故障问题进行评估，制定出一套或多套应急预案。

服务器是最主要的硬件设备。服务器要尽量使用专业的服务器，不要使用PC代替。专业的服务器中有多个CPU并且硬盘各方面的配置也比较优秀。由于服务器需要不停地工作，如果其中一个CPU或硬盘坏掉，别的CPU和硬盘还可以继续工作，不会影响到网站的正常运行。

2. 软件维护

软件维护主要包括操作系统、数据库及应用程序的维护。要及时给操作系统升级或者打上补丁，维护与升级各种服务器程序和Web应用程序，维护与升级安全防病毒软件等，以确保网站的24小时不间断地正常运行。

服务器操作系统配置是否可行以及配置是否优良，是确保一个网站长期良好运行的保证。除了要定期对操作系统进行维护外，还要定期对操作系统进行更新，使用最新稳定版本的操作系统。

动态网站要用到数据库，因此，需要维护数据库，使数据库保持运行在最佳状态。数据库维护主要包括：压缩数据库、清理废旧无用的数据、修复损坏的数据、检查系统空间占用、定期进行数据备份等。

应用程序的维护主要包括应用程序软件的安装、调试、运行、错误纠正、程序改进等。除了所开发的Web应用程序外，其他在服务器上安装的应用软件要尽可能地少，只要够用即可，这样可防止各个软件间的相互冲突。

3. 内容维护

网站内容维护的主要工作就是8.1节中介绍的更新网站的内容。这里不再赘述。

4. 安全维护

网站安全维护是指采取一定的安全手段，提高网站的安全性，主要包括网站安全管理、监测、数据备份、防止病毒的攻击和恶意的访问等。现在的黑客出于很多的商业目的，攻击网站是经常的事情，因此，网站安全维护是非常有必要的。一般情况下，在进行网站安全维护时可以从以下几方面入手。

(1) 安装补丁程序

操作系统是服务器运行的软件基础，其重要性不言而喻。但是，任何操作系统都是有漏洞的，进行网站维护时要及时地用补丁堵上漏洞。目前，许多企业的服务器操作系统使用的是微软的网络操作系统，主要有：Windows NT 4.0 Server、Windows 2000 Server以及Windows Server 2003等。

(2) 安装和设置防火墙

防火墙主要用于实现网络路由的安全性，在内部网与外部网之间的界面上构造一个保护层，并强制所有的连接都必须经过保护层，在此进行检查和连接。只有被授权的通信才能通过此保护层。因此，防火墙可以对计算机或网站提供保护，像盾牌一样屏蔽外界的恶意软件或不必要的网络流量的攻击。

(3) 安装入侵检测系统

入侵是指任何试图危害资源的完整性、保密性和可用性的活动。入侵检测就是检测入侵活动，并采取对抗措施。它会对未经授权的连接企图做出反应，甚至可以抵御一部分可能

的入侵。利用入侵检测系统的实时监控能力，可以发现正在进行的攻击行为及攻击前的试探行为，记录黑客的来源及攻击步骤和方法。

相对于传统意义的防火墙来说，入侵检测系统是一种主动防御系统，作为安全的一道屏障，可以在一定程度上预防和检测来自系统内外部的攻击。入侵检测系统处于防火墙之后，对网络活动进行实时检测，是防火墙的合理补充。

(4) 安装漏洞扫描工具

解决网络安全问题，首先要弄清网络中存在哪些安全隐患、脆弱点。漏洞扫描就是扫描网络系统、程序、软件的漏洞，是自动检测远端或本地主机安全脆弱点的技术。它查询端口，并记录目标的响应，收集关于某些特定项目的有用信息，如可用服务、拥有这些服务的用户、是否支持匿名登录、是否有某些网络服务需要鉴别等。

在服务器上安装能查找网络安全漏洞、评估并提出修改建议的商用或免费漏洞扫描工具，定期对服务器进行扫描，最大可能地发现潜在的安全问题，利用优化系统配置和打补丁等各种方式弥补最新的安全漏洞和消除安全隐患。

(5) 安装网络杀毒软件

计算机病毒是指编制或者在计算机程序中插入的破坏计算机功能或者毁坏数据、影响计算机使用，并能自我复制一组计算机指令或者程序代码。随着网络的发展和普及，计算机病毒的泛滥和危害十分严重，病毒的传播方式和破坏方式也越来越网络化。因此，需要安装高效的网络杀毒软件来控制病毒的传播。

目前，大多数反病毒厂商(如瑞星、江民、金山、卡巴斯基、赛门铁克、McAfee 等)都推出了网络版的杀毒软件，在选用某一杀毒软件时应综合考虑软件的价格、性能以及服务。同时，在网络版的杀毒软件使用中，必须要定期或及时升级杀毒软件。

(6) 账号和密码保护

账号和密码保护可以说是网站系统的第一道防线，目前网上的大部分对系统的攻击都是从截获或猜测密码开始的。一旦黑客进入了系统，那么前面的防卫措施几乎就没有作用，所以对服务器系统管理员的账号和密码进行管理是保证系统安全非常重要的措施。

网站系统要设置好账号和权限，设置的用户尽可能的少，对用户的权限尽可能的小，密码设置要足够强壮。系统管理员密码的位数一定要多，至少应该在 8 位以上，而且尽量将字母、数字及符号一起组合，不要设置成容易猜测的密码，如自己的名字、出生日期等。通过选择一个好的密码，并做好必要的保密工作，别人想要猜测或“破解”密码就非常困难。

(7) 监测系统日志

通过运行系统日志程序，系统会记录下所有用户使用系统的情形，包括最近登录的时间、使用的账号、进行的活动等。日志程序会定期生成报表，通过对报表进行分析，你可以知道是否有异常现象。

虽然开启日志服务对阻止黑客的入侵并没有直接的作用，但是可以通过它记录黑客的行踪，可以分析入侵者在我们的系统上到底做过什么手脚，给我们的系统到底造成了哪些破坏及隐患，黑客到底在我们的系统上留了什么样的后门，我们的服务器到底还存在哪些安全隐患等。

(8) 关闭不需要的服务和端口

服务器操作系统在安装的时候，会启动一些不需要的服务，这样会占用系统的资源，而



且也增加了系统的安全隐患。Windows 系统有很多默认的服务是没必要开的,甚至可以说是危险的,比如,默认的共享远程注册表访问,系统很多敏感的信息都是写在注册表里的。所以,应关闭不需要的服务。

网络端口可以为计算机与外界通信提供一个通道,但是也可以向黑客透露许多操作系统的敏感信息,例如,Windows 2000 Server 默认开启的 IIS 服务就告诉对方你的操作系统是 Windows 2000,69 端口告诉黑客你的操作系统极有可能是 Linux 或者 UNIX 系统。对端口的进一步访问,还可以返回该服务器上软件及其版本的一些信息,这些对黑客的入侵都提供了很大的帮助。同时,端口也可能成为黑客攻击计算机的一个入侵通道。因此,应该关闭那些不必要的端口。

(9) 定期对服务器进行备份

所谓备份,是指将网站数据以某种方式加以保留,以便在系统遭受破坏或其他特定情况下,重新加以利用的一个过程。在日常生活中,我们经常需要为自己家的房门多配几把钥匙,为自己的爱车预备一个备胎,这些都是备份思想的体现。为防止不能预料的系统故障,必须对系统进行安全备份。

除了对全系统进行每月一两次的备份外,还应经常对修改过的数据进行备份。在网站运行过程中,病毒的破坏、黑客的入侵、人为错误操作、人为恶意破坏、系统的不稳定、存储介质的损坏等等原因,都有可能造成重要数据的破坏。一旦数据出现丢失或者损坏,则可以利用备份进行数据恢复,及时地将系统恢复到正常状态。

11.2.1 任务描述

在“新闻类别网站”的运行过程中,维护网站使其正常运行。

11.2.2 任务实现

“新闻类别网站”的维护任务包括四个任务:

任务 1 对“新闻类别网站”进行日常硬件维护
任务 2 对“新闻类别网站”进行日常软件维护
任务 3 对“新闻类别网站”进行内容维护
任务 4 对“新闻类别网站”进行安全维护

任务 1:对“新闻类别网站”进行日常硬件维护

对“新闻类别网站”的网络设备进行日常维护和定期检修,定期完成硬件系统、电路等的维护测试,发现问题及时分析处理,保证硬件设备完好、系统运行正常。要注意网络机房室内的温度、湿度以及通风性,这些都将影响到服务器的散热和性能的正常运行。

任务 2:对“新闻类别网站”进行日常软件维护

对“新闻类别网站”的操作系统、数据库及相关应用程序进行日常维护。

任务 3:对“新闻类别网站”进行内容维护

对“新闻类别网站”的内容进行日常的更新维护。利用“新闻类别网站”的后台管理系统,可以发布、更新、删除新闻,如图 11-1 所示是新闻发布界面。



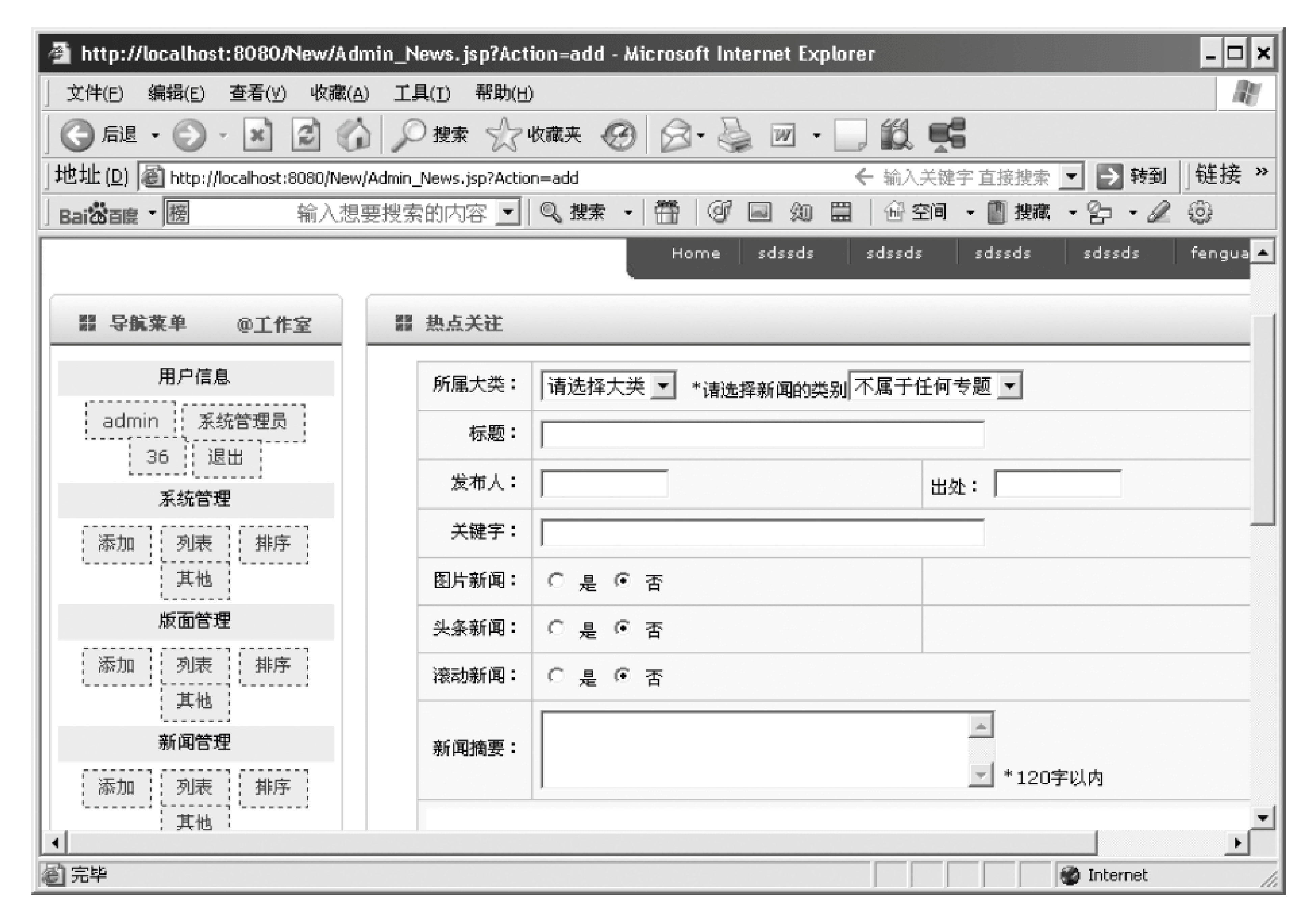


图 11-1　发布新闻

任务 4：对“新闻类别网站”进行安全维护

对“新闻类别网站”实施安全维护措施。

11.2.3　相关知识拓展

在目前常见的网站漏洞中，SQL 注入漏洞是最普遍也是最难防范的一个漏洞。注入漏洞的产生原因主要是程序员在编写代码的时候，没有严格过滤访问者提交的数据，使应用程序存在安全隐患。恶意访问者可以提交一段数据库查询代码，根据程序返回的结果，获得某些他想得知的数据，这就是所谓的 SQL Injection：SQL 注入漏洞（**注：**此部分内容在第 7.4 节的知识储备里介绍过，在那里主要考虑网站安全设计，这里主要考虑维护安全问题）。

1. SQL 注入漏洞的原理

SQL 注入漏洞的原理是恶意访问者从客户端提交特殊的代码，从而收集到程序及服务器的信息，获取想得到的资料。SQL 注入是从正常的 WWW 端口访问，而且表面看起来跟一般的 Web 页面访问没什么区别，所以目前的防火墙还不能对 SQL 注入发出警报提示。

SQL 注入漏洞是一种由于人为疏忽形成的漏洞，是网站程序员未对访问者从网页提交的非法数据进行过滤的结果。一个网站中需要对访问者提交的数据进行过滤的地方非常多，网页上的数据大都是从数据库中提取的，每个页面从数据库中读取了相应数据，然后将其数据放在页面上供访问者浏览。所以过滤访问者提交的数据是一项非常大的工程，稍不注意就有可能造成注入漏洞。



2. 防止和修复 SQL 注入漏洞的方法

可以通过一定的方法测试某个网页是否存在 SQL 注入漏洞,但是对于规模较大的网站,要找到有漏洞的页面工作量很大,可以利用 SQL 注入漏洞扫描工具,方便检测网站的 SQL 注入漏洞。

这里介绍一些防止和修复 SQL 注入漏洞的有效方法。

(1) SQL Server 的 sa 用户设置成强密码。

(2) Web 服务器与 DB(Database,数据库)服务器分别使用不同的机器。

(3) 随时注意相关软件是否有新的补丁需要补上。

(4) 修改网站文件中过滤数据的代码。

(5) 不要直接把代码运行中的错误信息显示到浏览器端。

(6) 修改数据库文件路径、数据库名称及网站后台登录路径。

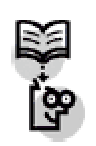
11.2.4 经验总结

网站维护是一项日常性的工作,专业性较强,所以一般应指派专人负责该项工作。与其他媒体一样,网站也是一个媒体,需要经常性的更新维护才会起到既定的商业效果。同时,为了使网站能安全、高效地运行,还需要进行硬件、软件及安全性维护,防范病毒感染、防止黑客攻击。安全维护的技术很多,各种技术都有其优缺点,需要结合实际的网络环境来选择最适合的综合技术方案。

对于微软的网络操作系统,因为使用的人特别多,所以暴露出来的 Bug 也特别多,同时,蓄意攻击它们的人也特别多。微软公司为了弥补操作系统的安全漏洞,在其网站上提供了许多补丁。可以从微软的网站上下载最新的 Service Pack(升级服务包),将安全漏洞及时补上。

对于防火墙,现在市场上有许多基于硬件或软件的防火墙,如瑞星、华为、神州数码、联想等厂商的产品。防火墙对于非法访问具有很好的预防作用,但是并不是安装了防火墙之后就万事大吉了,而是需要进行适当的设置才能起作用。如果对防火墙的设置不了解,需要请技术支持人员协助设置。

11.3 经营网站



知识储备

网站经营的好坏,是决定一个网站能否生存的关键。在网站经营过程中,要经常对网站进行优化,并且通过合理的方式实现赢利。

1. 网站优化

网站优化是通过对网站功能、网站结构、网页布局、网站内容等要素的合理设计,使网站内容和功能表现形式达到访问友好并易于宣传推广的最佳效果,从而充分发挥网站的网络营销价值。网站优化可分为网站的访问优化和网站的搜索引擎优化。

(1) 网站的访问优化

网站的主角是网站的访问者,访问者永远希望看到最新的内容,享受到更好的服务,因





此网站在运行阶段要经常进行网站的访问优化。

① 分析目标访问群体

在建站初期的需求分析中,需要分析目标访问群体,这一任务在网站经营中也非常必要。在经营过程中,要经常分析网站的流量来源,分析谁是目标访问群,目标访问群有什么特点,主要来自什么区域。要对访问网站的目的行为进行分析,提供所需要的服务,运用一系列推广方式来吸引目标群体加入。

② 更新网站信息

网站在发布之后,要对网站信息进行修改相对来说是很容易的,不像印刷品、CD、录像带以及其他发布形式,因此网站发布永远都没有最终版本。修改时,只需修改一个版本(即服务器上的版本)即可。对网站内容进行更新,简单的修改可以在几分钟内完成,而且访问者马上就能浏览。

③ 提高访问体验

访问体验是指访问者访问网站时的全部体验,包括对网站的满意度、对发现问题的忍受度,以及对网站期望与实际收益的差异程度。网站优化要以网站的有效性和实用性为目标,提高访问体验。要经常与访问者进行沟通交流,关注访问者的反馈。

对网站的修改不必局限于已有内容的细化,也可以根据用户的需求增加新的功能,还可以随着客户业务的增长和新产品的上市而增加新的内容。这些修改对于网站特色是非常重要的,对于提高用户体验也是非常有作用的。

(2) 网站的搜索引擎优化

搜索引擎是 Internet 用户最常用的网络工具,网站在搜索引擎上的排名对于网站访问量有巨大的影响力。如果网站的搜索引擎排名位于 50 名之后,点击率基本上很小。网站的搜索引擎优化是从搜索引擎的分析角度对网站架构、布局、链接、关键字、内容等方面实施整体优化,让搜索引擎更好地收录企业网站页面,在搜索结果页面中获得更好的、稳定的排名。

① 关键字分析

关于关键字的选择,在 10.3.3 小节中已经介绍过,这里再强调一下,关键字很重要。关键字应该是描述网站的产品及服务的词语,而且要选取那些人们在搜索时所常用到的关键字。因此,一定要了解关注网站的客户需求,分析他们会用什么样的关键字来找到网站,掌握这些关键字以后,在网页的标题和描述中加以体现,那么对网站的搜索引擎排名是大有好处的。

② 网页设计优化

网页的设计对于搜索引擎搜索网站也是十分重要的。网页内容要完整、充实并且经常更新。网站页面要尽量简洁,这样可以加快下载速度。选择一个信誉度高的服务商,保证服务器的响应速度。对网页上的图片也要优化,在网页制作时,对图片要加入文字注释的 Alt 标签,而且文字注释中应尽量放入关键字。

③ 增加外部链接

外部链接是搜索引擎判断一个网站是否优秀、是否重要的一个重要因素。当有很多网站主动链接一个网站时,搜索引擎会认为哪个网站很重要,给予的权重就会很高。外部链接的数量和质量成为一个网站的重要评估标准,所以外部链接对搜索引擎优化的重要性就显而易见了。

④ 搜索引擎竞价排名

搜索引擎为了赢利,现在都提供竞价排名。当然,也可以免费登录搜索引擎,但排名肯



定不会在前面，因此很多企业会做搜索引擎上的竞价排名。目前国内用得较多的搜索引擎是百度、中文 Google、中文雅虎等。

2. 网站赢利

网站赢利是指通过经营网站来赚钱，直接获得经济利益。当前互联网网站的赢利模式有以下几种。

(1) 在线广告收费

在线广告是比较普遍的网站赢利方式，其广告形式多种多样，有 Banner 广告、Logo 广告、Flash 多媒体动画、在线影视等。当网站拥有较可观的流量时，就具备了网站广告收费的条件，可以通过在网页放置广告收取一定的费用来增加收入。

从收费的方式来看，现在经常用的是按点击次数收费，Google 和百度等搜索引擎网站都是主要采取此类广告方式。具体可以通过互联网广告公司，如 Google Adsense，或者通过联系实体企业，让企业主为其投放广告。第一种方式操作起来简单，被多数中小型个人网站采用；第二种实现起来需要有一定的影响力或社会关系，网站内容也要符合企业产品的要求，但收入比第一种可观。

(2) 网上零售(B2C)

网上零售主要是指个人通过 Internet 购买商品或享受服务。购物者可以浏览网上的商品目录，选择满意的商品或服务，通过网上下订单、网上付款或离线付款、卖方处理订单、网上送货或离线送货，完成整个购物过程。

目前开发网上零售业务的 B2C 零售网站有许多，比如著名的有亚马逊、当当网、卓越网等，各大网站也都有自己的网上商城。通常的 B2C 网上零售有两种操作方法，一种是自己经销产品，通过互联网销售；另一种则是建立一个网上零售的平台，让更多的商家通过此平台销售他们的产品。

(3) 部分信息收费或收取会员费

在上网时，经常会遇到某些网站要求访问者必须成为注册用户，才能阅读此网站提供的信息，有些网站甚至要求访问者必须是收费用户，这就是所谓的网站特殊信息收费服务。例如，一些企业需要的专业技术信息、个人需要的人才招聘信息、有价值的电子图书、流行的在线电影、人气旺的交友网站等可能就只提供收费访问。

当网站提供质量较高的内容或服务时，可以通过这种方式实现收入。此种方式需要先有一定吸引力的免费内容或服务，以此为基础积累访问用户基数，通过提供更具吸引力的增值服务来顺利过渡至收费。

(4) 软件下载

软件下载可以认为是网上零售的一部分，只是软件是一种特殊产品，可以在线直接下载，无须物流的运输过程。

互联网上的软件下载有的是免费形式，有的是收费形式，一般根据所下载的软件而定。人们一般会选择下载免费软件。当然，一些比较专业且价格不菲的正版软件的下载通常是收费的，即使是共享也是有使用期限的。有许多的软件公司更是利用互联网的优势进行在线升级服务，这也是促进与网站访问群体互动的好方法。

(5) 短信铃声等移动运营服务

近几年兴起的互联网短信铃声下载，不仅为手机用户带来了更周到的服务和更精彩的



铃声彩信,也为各大网站提供了一个非常良好的人气来赚取利润的赢利模式。如果网站拥有较好的流量,可以通过与专业短信铃声的服务提供商网站进行合作,获得一些赢利。

上述模式各有特点,应该说网上零售模式是最长久也最有前途的赢利模式,将是未来互联网发展的一个重要分支方向。

11.3.1　任务描述

在“新闻类别网站”的运行过程中,及时优化网站,尽可能快地使其走向赢利。

11.3.2　任务实现

对于“新闻类别网站”的赢利,因为其主要内容产品是新闻与资讯,可以考虑的赢利思路有两种:一种思路是通过网站内容吸引访问者的注意力,加大网站的访问量,然后通过在线广告收费获得利润;另一种思路是提供给访问者需要的内容,通过内容直接收费。不管是哪种方式,都意味着网站要有较好的流量,网站内容要有较高的阅读价值。

11.3.3　相关知识拓展

关于网站赢利,人们经常会提到电子商务。电子商务是对整个贸易活动的电子化,通常指通过 Internet 进行的商务活动,包括广告、交易、支付、服务等活动。很多的赢利活动是通过电子商务网站进行的。

根据电子商务交易主体的不同,电子商务可以分为以下 5 类。

(1) 企业与消费者之间

企业与消费者之间(Business to Consumer,B2C)的电子商务基本上可以描述为在线购物和获取服务,它与人们的生活关系最为密切,这是人们最为熟悉的一种模式。通过网上商店,企业可以更方便、更全面、更经济地为消费者提供产品信息。这种模式的代表有卓越亚马孙书店(http://www.amazon.com)、当当网(http://www.dangdang.com)等。

(2) 企业与企业之间

企业与企业之间(Business to Business,B2B)的电子商务是指企业与企业之间通过互联网进行产品、服务及信息的交换,通过 B2B 的交易方式,买卖双方能够在网上完成整个业务流程。B2B 使企业之间的交易减少许多事务性的工作流程和管理费用,降低了企业经营成本,而且网络的便利及延伸性使企业扩大了活动范围,企业发展跨地区跨国经营更方便,成本更低廉。这种模式的代表是阿里巴巴网(http://china.alibaba.com)。

(3) 消费者与消费者之间

消费者与消费者之间(Consumer to Consumer,C2C)的电子商务是指个人与个人之间通过 Internet 等媒体进行的交易活动。由于 Internet 的日益普及,以及许多支持 C2C 交易的网站的推广,使得这类电子商务也很常见。在拍卖行业,比较典型的是国内的淘宝网(http://www.taobao.com),支持 C2C 交易,集个人竞价、集体议价、标价求购三种交易模式于一体,提供包括买卖信息、在线交易和电子支付等在内的全套解决方案。



(4) 企业与政府之间

企业与政府之间(Business to Government,B2G)的电子商务是指电子商务活动覆盖企业与政府组织间的各项事务,目前我国有些地方已经推行网站采购,政府采购的细节在Internet上公布,通过网上竞价方式进行招标,企业也通过电子的方式进行投标。我国的金关工程是通过B2G电子商务,建立我国以外贸为龙头的电子商务框架,并促进我国各类电子商务活动的开展。

(5) 消费者与政府之间

消费者与政府之间(Consumer to Government,C2G)的电子商务是指政府利用电子商务扩展到进行福利费发放和个人所得税征收等方面的工作。例如,网上办税服务大厅的服务,虽然这类服务还没有达到真正的报税电子化,但是,它已经具备了消费者与政府机构之间电子商务的雏形。

电子商务能将传统的商务流程数字化、电子化,突破时间空间的局限,从而大大提高商业运作的效率,并有效地降低了成本。作为一种全新的商业模式,电子商务将日益深入人心,而电子商务网站作为电子商务的运作平台,也必将迎来更大的发展机遇。

11.3.4 经验总结

网站的真正价值体现在运营阶段,因此网站的经营很关键。但是网站运营维护也不能抱着立竿见影的思想。从长期来看,良好的运营维护是一定会产生好的效果的,已经有越来越多传统行业的企业网站对销售业绩的增长产生较大的作用。

网站优化主要是两个方面:一方面针对网站的访问者,改善访问者的访问体验,如加快网站访问速度、使网站内容更有价值、网站栏目路径设置更合理等;另一方面是针对搜索引擎,改善搜索引擎友好性,提高搜索引擎排名。

尽量增加其他网站指向自己网站的链接数目。常用的增加网站外部链接的方法有以下几种。

- 友情链接:尽量和与自己内容相关的网站交换友情链接。
- 登录分类目录:把网站提交给分类目录,其操作方法和登录搜索引擎门户网站差不多。
- 登录导航网址:很多人喜欢用hao123、265等一些网址大全网站,如果网站可以被大型导航网址网站收录,无疑会给网站带来不少被点击的机会。

网站要想在搜索引擎中获得好的排名,竞价价格是一个关键因素一般情况,出价越高,排名越靠前。还有一个因素,就是网站的质量,包括关键字、页面内容、更新程度、页面结构等因素。例如,百度的排名是由综合排名指数决定的,综合排名指数=质量度×竞价价格。所以,如果网站的质量度较高,则也可以用稍低的价格获得靠前的排名。

11.4 实　　训

参照本章的工作步骤,更新、维护“新闻类别网站”,并尽可能使该网站迈进赢利的门槛。



参 考 文 献

[1] 杨选辉.网页设计与制作教程.北京：清华大学出版社，2005

[2] 彭雪冬，柯建林，吕洋波.网站建设实用开发精粹.北京：人民邮电出版社，2006

[3] (英)Andy Budd，Cameron Moll，Simon Collison.精通 CSS——高级 Web 标准解决方案.陈剑瓯译.北京：人民邮电出版社，2006

[4] 万峰科技.JSP 网站开发四"酷"全书——新闻、论坛、电子商城、博客/网站开发专家.北京：电子工业出版社，2005

[5] 杨磊，陈凌云.精通 Eclipse Web 开发——Java 体系结构、工具、框架及整合应用.北京：人民邮电出版社，2006

[6] 赵生慧.Java 面向对象程序设计.北京：高等教育出版社，2007

[7] 邓子云，郝斌.JSP 应用开发.北京：机械工业出版社，2008

[8] 杜文浩，景秀丽.软件测试基础教程.北京：中国水利水电出版社，2008

[9] 蓝色理想网站 http://www.blueidea.com/

[10] 52CSS 网站 http://www.52css.com

[11] 网页设计师联盟网站 http://www.68design.net/

[12] 网页教学网网站 http://www.webjx.com/